Journal of Intelligent and Robotic Systems **28:** 195–211, 2000.

A Genetic Approach for Simultaneous Design of Membership Functions and Fuzzy Control Rules

CHIA-JU WU
Department of Electrical Engineering, National Yunlin University of Science and Technology, Touliu, Yunlin 640, Taiwan, R.O.C.; e-mail: wucj@pine. yuntech.edu.tw

GUAN-YU LIU
Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Touliu, Yunlin 640, Taiwan, R.O.C.

(Received: 8 June 1999; in final form: 4 October 1999)

Abstract. Based on the genetic algorithm (GA), an approach is proposed for simultaneous design of membership functions and fuzzy control rules since these two components are interdependent in designing a fuzzy logic controller (FLC). With triangular membership functions, the left and right widths of these functions, the locations of their peaks, and the fuzzy control rules corresponding to every possible combination of input linguistic variables are chosen as parameters to be optimized. By using a proportional scaling method, these parameters are then transformed into real-coded chromosomes, over which the offspring are generated by rank-based reproduction, convex crossover, and nonuniform mutation. Meanwhile, the concept of enlarged sampling space is used to expedite the convergence of the evolutionary process. To show the feasibility and validity of the proposed method, a cart-centering example will be given. The simulation results will show that the designed FLC can drive the cart system from any given initial state to the desired final state even when the cart mass varies within a wide range.

Key words: fuzzy logic controllers, membership functions, fuzzy control rules, genetic algorithms.

1. Introduction

Fuzzy logic control is based on the concept of the fuzzy algorithm [10, 16]. The typical architecture of an FLC is shown in Figure 1, which is comprised of four principal components: a fuzzification interface, a knowledge base, a decision-making logic, and a defuzzification interface [9]. The fuzzification interface transforms crisp measured data into suitable linguistic values. The knowledge base consists of a database and a fuzzy control rule base, in which the former provides necessary definitions to define linguistic control rules and fuzzy data manipulation, and the latter is characterized by a set of linguistic control rules. The decision-making logic has the capability of simulating human decision-making based on fuzzy concepts and of inferring fuzzy control actions employing fuzzy implication and the rules of inference in fuzzy logic. The defuzzification interface is utilized to yield a nonfuzzy control action from an inferred fuzzy control action.

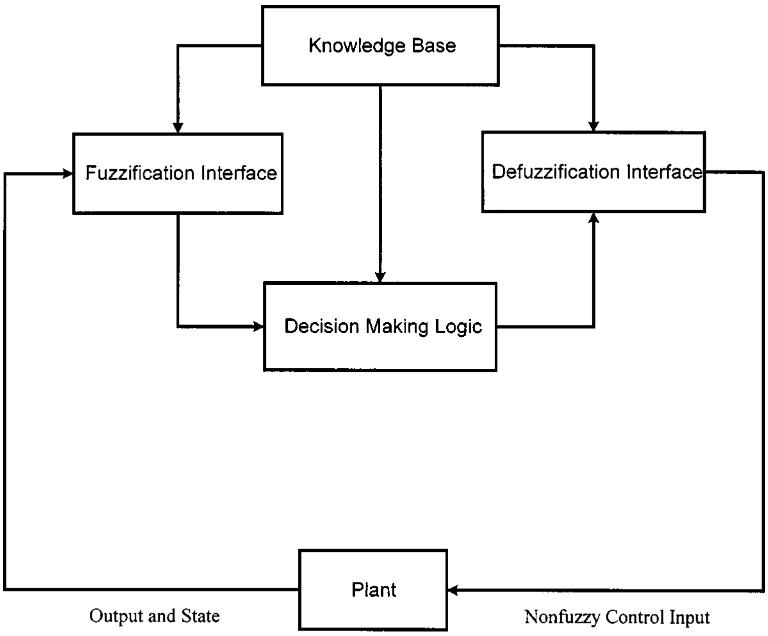


Figure 1. Typical architecture of an FLC.

In designing an FLC, membership functions and fuzzy control rules are usually determined serially in most of the previous works. For example, the membership functions are chosen first and then used in the design of fuzzy control rules. In this manner, the fuzzy control rules will be optimal for the specific membership functions only. If different membership functions are chosen, then it is possible that another rule set will be more appropriate. This in turn motivates the study in this paper and GAs are adopted since they provide a means to search poorly understood, highly complex spaces.

Several researchers have used GAs in the design of FLCs previously [1, 7, 8, 14, 15]. In the works of [1, 8, 15], however, the focus is on the development of the membership functions or the fuzzy control rules only, not both together. In this case, the result obtained is a hand-designed rule set with GA-designed membership functions or hand-designed membership functions with a GA-designed rule set. Obviously, this does not use GAs to the full advantage.

Membership functions and fuzzy control rules are designed simultaneously in [7, 14]. In the work of [14], the chromosome consists of two types of genes, the control genes and the parameter genes. The control genes, in the form of bits, determine the membership function activation, whereas the parameter genes are in the form of real number to represent the membership functions. These two kinds of genes are arranged in a hierarchical form and one type of gene controls the other type of gene. Therefore, this method is called the Hierarchical Genetic Algorithm (HGA). It is mentioned in [14] that the HGA has the ability to reach an optimal set of memberships and rules without a known overall fuzzy set topology. However, no mathematical proof can be found and only the example of a constant water pressure pumping system is used for illustration. Moreover, one also cannot find the comparison between the HGA and any other GA scheme.

The FLC in [7] is characterized by a collection of fuzzy IF–THEN rules, which take the form of Mamdani implication [10]. Triangular membership functions are used for the input and output membership functions of the FLC. The base lengths of the input membership functions and the fuzzy rules are chosen as parameters to be optimized simultaneously. When applying the Simple Genetic Algorithm [4] to determine these parameters, binary strings are used for encoding.

Though the effectiveness of the FLC in [7] has been verified through simulation examples, several disadvantages still exist. For example, it is required in [7] that the triangular membership functions must be symmetric and the locations of the peaks of these functions are fixed and not adjustable. It is obvious that these constraints will reduce the flexibility in designing FLCs. Therefore, an approach is proposed in this paper to cope with this problem. In the proposed method, the left and right widths of membership functions, the locations of their peaks, and the fuzzy control rules are all chosen as parameters to be optimized simultaneously. In the evolutionary process, genetic operators, such as real number encoding, convex crossover, and nonuniform mutation, will be implemented [3, 12]. Moreover, the concepts of enlarged sampling space [2, 13] and rank-based fitness [3] will also be used to expedite the convergence of the evolutionary process. For illustration and comparison, the cart-centering problem in [7, 15] will be included in the simulation example. The objective is to design an FLC that can determine the force to bring the cart from an arbitrary initial state to the state of zero velocity and zero location in minimum time. For the designed FLC to be able to operate over the entire range of the input space, multiple initial states will be taken into account in defining the fitness function. In this manner, the proposed method will be fully capable of creating a complete fuzzy logic controller provided that the state equations of the controlled plant are given. This will eliminate the need for human expertise in the design procedure.

2. Fuzzy Control Rules and Membership Functions

Fuzzy control rules are a collection of fuzzy IF {condition} THEN {action} rules, which characterize the input–output relation of the system. Since a multi-input multi-output (MIMO) system can always be decomposed into a set of multi-input single-output (MISO) systems, only the case of MISO systems will be considered. Moreover, since the simulation example that will be discussed later has two inputs and one output, the two-input single-output (TISO) case will be used for illustration without any loss of generality.

Let a, b, and c denote the input and output linguistic variables of the TISO system, and $A_1, A_2, \ldots, A_{n_1}$, $B_1, B_2, \ldots, B_{n_2}$, and $C_1, C_2, \ldots, C_{n_3}$ be the linguistic

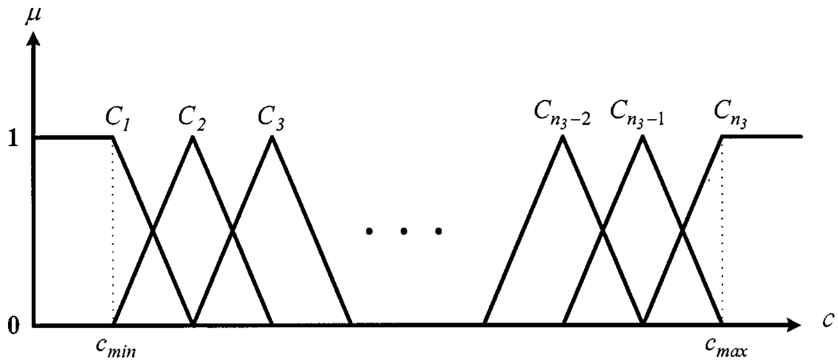


Figure 2. Symmetric membership functions of the output linguistic variable.

values that a, b, and c may take, respectively. In this case, the FLC of the TISO system will be expressed as

$$\begin{array}{l}
\text{IF } a \text{ is } A_1 \text{ AND } b \text{ is } B_1, \text{ THEN } c \text{ is } C_{11} \\
\qquad\qquad\qquad\vdots \\
\text{IF } a \text{ is } A_1 \text{ AND } b \text{ is } B_{n_2}, \text{ THEN } c \text{ is } C_{1n_2} \\
\text{IF } a \text{ is } A_2 \text{ AND } b \text{ is } B_1, \text{ THEN } c \text{ is } C_{21} \\
\qquad\qquad\qquad\vdots \\
\text{IF } a \text{ is } A_2 \text{ AND } b \text{ is } B_{n_2}, \text{ THEN } c \text{ is } C_{2n_2} \\
\qquad\qquad\qquad\vdots \\
\text{IF } a \text{ is } A_{n_1} \text{ AND } b \text{ is } B_{n_2}, \text{ THEN } c \text{ is } C_{n_1 n_2},
\end{array} \tag{1}$$

where $C_{ij} \in \{C_1, C_2, \ldots, C_{n_3}\}$ for $1 \leqslant i \leqslant n_1,\ 1 \leqslant j \leqslant n_2$.

The input and output membership functions of an FLC are usually parametric functions such as triangular functions, trapezoidal functions, and bell-shaped functions. Though the proposed method is equally applicable to all these three kinds of membership functions, the triangular ones will be used in this paper. In defining the membership functions for $C_1, C_2, \ldots, C_{n_3}$, for simplification of the defuzzification process, these functions will be assumed to be symmetric and evenly distributed as shown in Figure 2, where $[c_{\min}, c_{\max}]$ is the actual physical domain over which c takes its crisp values. For each individual membership function of $A_1, A_2, \ldots, A_{n_1}$ and $B_1, B_2, \ldots, B_{n_2}$, the left width, the right width, and the location of the peak will be chosen as parameters to be optimized. The illustration for these parameters is shown in Figure 3.

3. Genetic Algorithms

From the above description, one can find that the main objective of this paper is to design the membership functions $A_1, A_2, \ldots, A_{n_1},\ B_1, B_2, \ldots, B_{n_2}$, and the fuzzy control rules $C_{ij},\ 1 \leqslant i \leqslant n_1,\ 1 \leqslant j \leqslant n_2$, simultaneously based on a

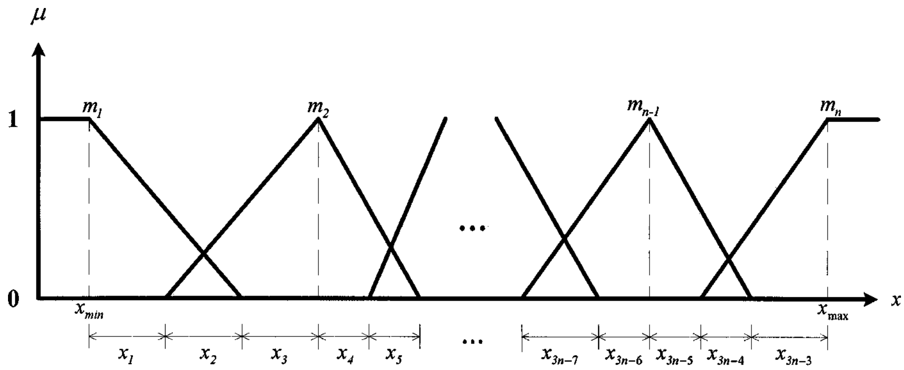


Figure 3. A set of n membership functions, in which there are $(3n - 3)$ parameters to be determined.

chosen criterion. Obviously, this is not an easy task since many parameters must be determined at the same time. This, in turn, explains why GAs are adopted since they provide a means to search poorly understood, highly complex spaces. However, before going further, several issues on GAs must be discussed first.

3.1. CHROMOSOME REPRESENTATIONS

How to encode a solution of the problem into a chromosome is a key issue for GAs. In the past, bit string encoding [6] was the most popular approach used by GA researchers because of its simplicity. However, for many GA applications, especially for the problems from the industrial engineering world, the binary string is difficult to apply directly because it is not a natural coding. In this paper, since the parameters to be determined are all real, a real number representation will be used, in which each chromosome vector will be coded as a vector of real numbers of the same length as the solution vector.

Once the real-coded chromosomes are used, the next step is to determine the number of genes in a chromosome. From the illustration in Figure 3, one can find that there are $(3n_1 - 3)$ and $(3n_2 - 3)$ parameters to be determined for membership functions of $A_1, A_2, \ldots, A_{n_1}$ and $B_1, B_2, \ldots, B_{n_2}$, respectively. Moreover, to determine fuzzy control rules $C_{ij},\ 1 \leqslant i \leqslant n_1,\ 1 \leqslant j \leqslant n_2,\ (n_1 \cdot n_2)$ parameters will be needed. Therefore, one can find easily that a chromosome will contain $(3n_1 + 3n_2 + n_1n_2 - 6)$ genes. The first $(3n_1 - 3)$ genes, the next $(3n_2 - 3)$ genes, and the last (n_1n_2) genes will be used to determine the membership functions of $A_1, A_2, \ldots, A_{n_1}$, the membership functions of $B_1, B_2, \ldots, B_{n_2}$, and the control rules $C_{ij},\ 1 \leqslant i \leqslant n_1,\ 1 \leqslant j \leqslant n_2$, respectively.

In determining the range for each gene of a chromosome

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_{3n_1+3n_2+n_1n_2-6}],$$

since

$$C_{ij} \in \{C_1, C_2, \ldots, C_{n_3}\} \quad \text{for } 1 \leqslant i \leqslant n_1,\ 1 \leqslant j \leqslant n_2,$$

one can find that $x_{3n_1+3n_2-5}, x_{3n_1+3n_2-4}, \ldots, x_{3n_1+3n_2+n_1n_2-6}$ are all within the range $[1, n_3]$. The values of 1 and n_3 will correspond to C_1 and C_2, respectively. For $x_1, x_2, \ldots, x_{3n_1+3n_2-6}$, however, their corresponding ranges are not easy to be determined since the following two equations must be satisfied at every time instant as illustrated in Figure 3:

$$\sum_{i=1}^{3n_1-3} x_i = a_{\max} - a_{\min}, \tag{2}$$

$$\sum_{i=3n_1-2}^{3n_1+3n_2-6} x_i = b_{\max} - b_{\min}, \tag{3}$$

where $[a_{\min}, a_{\max}]$ and $[b_{\min}, b_{\max}]$ correspond to $[x_{\min}, x_{\max}]$ in Figure 3, and are the actual physical domains over which the input linguistic variables a and b take their crisp values, respectively.

Since $x_1, x_2, \ldots, x_{3n_1+3n_2-6}$ are real numbers to be adjusted by GAs, their values will change from one generation to another in the evolutionary process. In this case, it is very possible that the constraints in (2) and (3) will be violated. Therefore, a proportional scaling method will be used to solve this problem, in which the values of the $(3n_1 + 3n_2 - 6)$ genes are determined according to their proportions, not from the values themselves. This means the value of each gene is determined as follows:

$$x_i' = (a_{\max} - a_{\min}) \frac{x_i}{\sum_{i=1}^{3n_1-3} x_i} \qquad \text{for } 1 \leqslant i \leqslant 3n_1 - 3, \tag{4}$$

$$x_i' = (b_{\max} - b_{\min}) \frac{x_i}{\sum_{i=3n_1-2}^{3n_1+3n_2-6} x_i} \qquad \text{for } 3n_1 - 2 \leqslant i \leqslant 3n_1 + 3n_2 - 6. \tag{5}$$

In this manner, since the constraints in (2) and (3) will always be satisfied, the ranges of $x_1, x_2, \ldots, x_{3n_1+3n_2-6}$ can be chosen almost arbitrarily. However, if only two neighboring membership functions are assumed to have intersection as shown in Figure 3, then the lower bounds of the ranges must be chosen to be greater than zero.

3.2. CROSSOVER AND MUTATION OPERATIONS

For real-coded chromosomes, there exist different kinds of crossover operators such as the linear crossover, the affine crossover, and the convex crossover [3, 12]. Among them, the convex crossover may be the most commonly used one and is defined as follows for two real-coded chromosomes $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

$$\boldsymbol{x}_1' = \lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2, \tag{6}$$

$$\boldsymbol{x}_2' = \lambda \boldsymbol{x}_2 + (1 - \lambda)\boldsymbol{x}_1, \tag{7}$$

where $\lambda \in (0, 1)$.

The basic concept of convex crossover is borrowed from the convex set theory and it has the property of closure for a convex solution set. This means that for any two chromosomes $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in a solution space D, the chromosomes $\boldsymbol{x}_1'$ and $\boldsymbol{x}_2'$ generated by (6) and (7) will also be in D provided that D is a convex set.

Similarly, there also exist different mutation operators such as the uniform mutation and the nonuniform mutation for real-coded chromosomes. The difference between these two operators is that the action of nonuniform mutation depends on the age of the population while the uniform operation has no such a property. Therefore, the nonuniform mutation operator is adopted in this paper. For a given parent $\boldsymbol{x}$, if a gene x_k of it is selected for nonuniform mutation, then the resulting offspring will be randomly selected from one of the following two choices:

$$x_k' = x_k + \left(x_k^U - x_k\right) r \left(1 - \frac{g}{G}\right)^b, \tag{8}$$

$$x_k' = x_k - \left(x_k - x_k^L\right) r \left(1 - \frac{g}{G}\right)^b, \tag{9}$$

where $[x_k^L, x_k^U]$ is the range of x_k, r is a random number from $[0, 1]$, g is the generation number, G is the maximal generation number, and b is a parameter determining the degree of nonuniformity.

From the operations in (8) and (9), one can find that the function $(1 - (g/G))^b$ returns a value in the range $[0, 1]$ such that the value $(1 - (g/G))^b$ approaches to 0 as g increases. This property causes this operator to search the solution space uniformly initially (when g is small), and locally at later stages.

3.3. ENLARGED SAMPLING SPACE

The principle behind GA is essentially Darwinian natural selection. Therefore, selection provides the driving force in a GA and the selection procedure may create a new population for the next generation based on either all parents and offspring or part of them. The regular sampling space [6] contains all offspring but just part of parents since parents are replaced by their offspring soon after they give birth. In this manner, since genetic operators are blind in nature, it is possible that the offspring will be worse than their parents and some fitter chromosomes will be lost in the evolutionary process. Therefore, to cope with this problem, the selection procedure in this paper will be performed on enlarged sampling space [3], which contains whole of parents and offspring and both parents and offspring have the same chance of competing for survival. An evident advantage of this approach is that one can improve the performance of GAs by increasing the crossover and mutation rates since high rates will not introduce too much random perturbation if selection is performed on enlarged sampling space [3].

3.4. RANKING MECHANISM

When performing selection, the selection probability of a chromosome is usually proportional to its fitness. However, this scheme exhibits some undesirable properties such as a few super chromosomes will dominate the selection process in early generations. Moreover, competition among chromosomes will be weaker and a random search behavior will emerge in later generations. Therefore, the ranking mechanism is used in this paper to mitigate these problems, in which the chromosomes are selected proportionally to their ranks rather than actual evaluation values. This means that the fitness will be an integer number from 1 to N, where N is the population size. The best chromosome will have a fitness value equal to N and the worst one will have a fitness value equal to 1.

3.5. PROBLEM SOLUTION

The details of the proposed method can be summarized as follows:

ALGORITHM A.

Step 1: Given the membership functions of the output linguistic variable.
Step 2: Define the fitness function.
Step 3: Determine the population size, the crossover rate, and the mutation rate.
Step 4: Produce an initial generation in a random way.
Step 5: Evaluate the fitness for each member of the generation.
Step 6: With the crossover rate in Step 3, generate offspring according to (6) and (7), in which the ranking mechanism is used for selection of chromosomes.
Step 7: With the mutation rate in Step 3, generate offspring according to (8) and (9).
Step 8: Select the members of the new generation from the parents in the old generation and the offspring in Step 6 and Step 7 according to their fitness values.
Step 9: Repeat the procedure in Step 6 through Step 8 until the number of generations reaches a prescribed value.

4. Simulation Results

The cart-centering problem in [7, 15] will be used as the simulation example, in which the state equations of the cart are described as follows:

$$p(t+\Delta t) = p(t) + \Delta t \cdot v(t), \tag{10}$$

$$v(t+\Delta t) = v(t) + \Delta t \frac{F(t)}{m}, \tag{11}$$

where Δt is the sampling period, $F(t)$ is the input force, $m,\ p(t)$, and $v(t)$ are the mass, the position, and the velocity of the cart, respectively. Meanwhile, the following constraints will be added throughout the simulation:

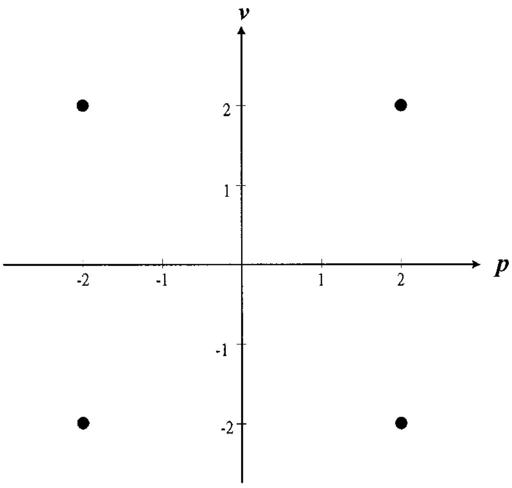
$$\begin{aligned} m &= 20 \text{ kg}, && (12)\\ \Delta t &= 0.02 \text{ s}, && (13)\\ -2 \text{ m} &\leqslant p(t) \leqslant 2 \text{ m}, && (14)\\ -2 \text{ m/s} &\leqslant v(t) \leqslant 2 \text{ m/s}, && (15)\\ -150 \text{ N} &\leqslant F(t) \leqslant 150 \text{ N}. && (16) \end{aligned}$$

The objective is to design an FLC which can determine the force $F(t)$ that will drive the cart from an arbitrary initial condition to the desired final condition $(p, v) = (0, 0)$ in the time-optimal manner. Since the choice of initial condition is arbitrary, the strategy of multiple initial conditions will be used in defining the fitness function. That is to choose a proper set of initial conditions and define the fitness function as the sum of the fitness of each individual initial condition. The ideal case is to take all points in the input space into account. However, this is impossible to do since there are an infinite number of initial conditions to be chosen. Meanwhile, more initial conditions also lead to longer computation time. Therefore, two cases will be used for illustration and comparison, in which four and sixteen initial conditions in the input space are chosen, respectively, as shown in Figure 4. In this manner, assume that $200\Delta t$ is the limit of time steps in the simulation, one possible definition of the fitness function will be

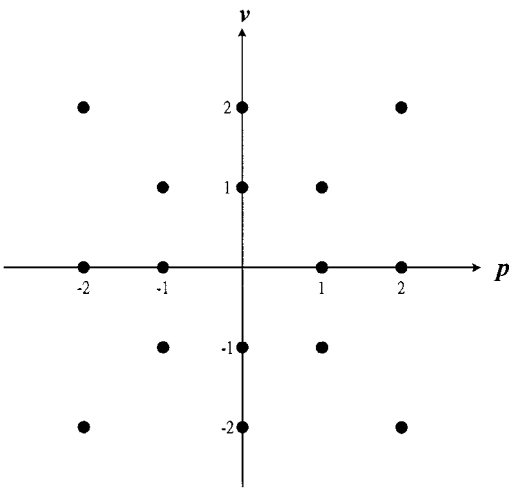
$$\text{fitness} = \sum_{i=1}^{n}(200 - T_i) \cdot \min(T_1, T_2, \ldots, T_n), \tag{17}$$

where n is the number of initial conditions, T_i is the time steps needed for each initial condition to reach the desired final condition, and $\min(T_1, T_2, \ldots, T_n)$ is the minimum value among $T_1, T_2, \ldots, T_n$. From this definition, if the fitness value is found to be zero, then one can conclude that at least one of the n initial conditions cannot be driven to the desired final condition.

In applying the GA, referring to the works [5, 11], the crossover rate, the mutation rate, the population size, and the maximal generation number are chosen to be 0.8, 0.2, 50, and 100, respectively. Meanwhile, it will be assumed that there are five membership functions for each of the input and output linguistic variables of the designed FLC. In this case, as illustrated in Section 3.1, each chromosome will contain 49 ($= 3 \cdot 5 + 3 \cdot 5 + 5 \cdot 5 - 6$) genes, in which the first 24 genes will determine the membership functions of the input variables and the last 25 genes will determine the fuzzy control rules. Moreover, the ranges of the first 24 genes will all be chosen as $[1, 100]$, and those of the last 25 genes will all be chosen as $[1, 5]$, respectively. In performing the mutation operation in (8) and (9), the degree of nonuniformity is chosen to be 2.



(a)



(b)

Figure 4. Different initial conditions in the input space: (a) four initial conditions; (b) sixteen initial conditions.

4.1. THE CASE OF FOUR INITIAL CONDITIONS

Applying the proposed method to the cart-centering problem with the four initial conditions shown in Figure 4(a), the plot of the fitness value is obtained as shown in Figure 5. From this figure, one can find that the fitness value increases monotonically from 56156 to 61744 in 100 generations since the enlarged sampling space is used. The optimal membership functions for the two input linguistic variables p and v, and the optimal fuzzy control rules to generate F are determined as shown in Figures 6 and 7, respectively.

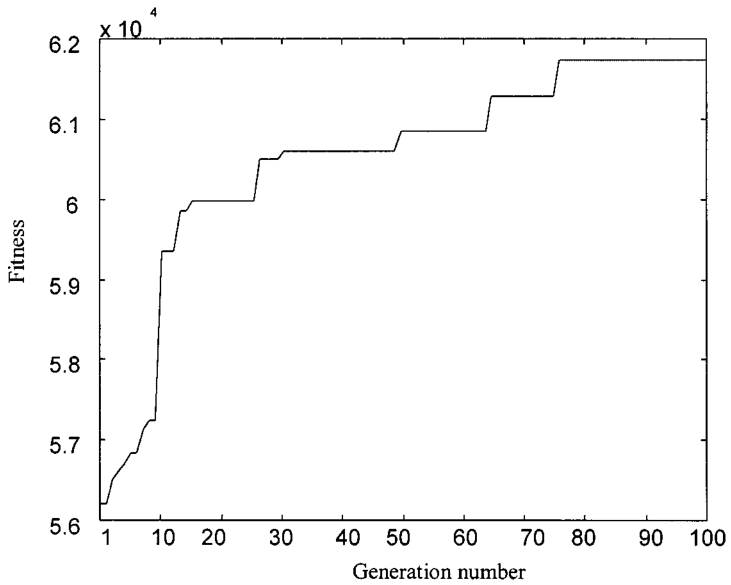


Figure 5. Plot of fitness value in each generation.

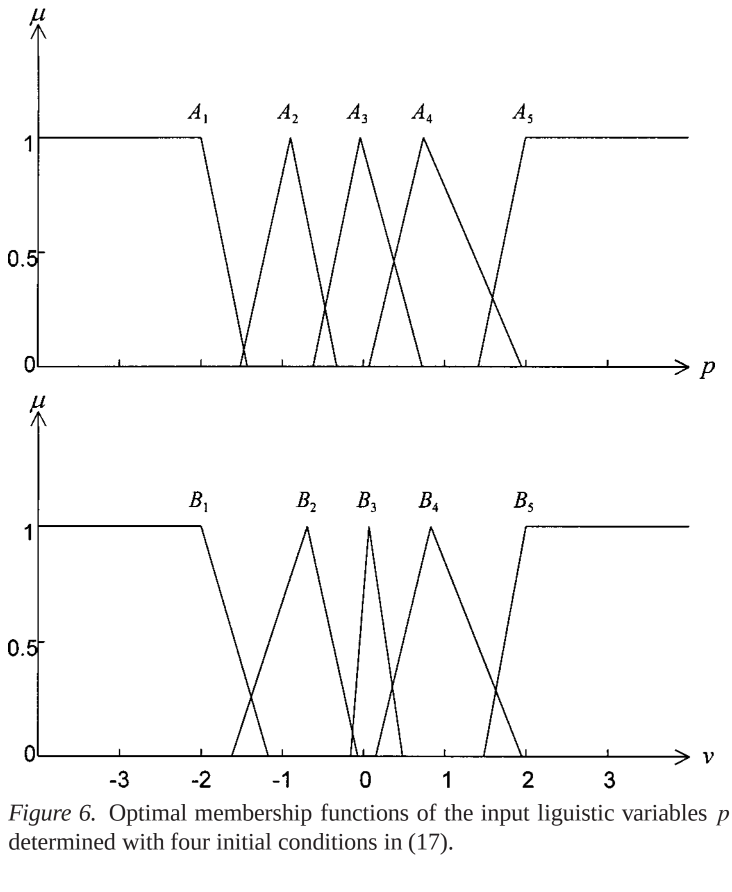


Figure 6. Optimal membership functions of the input liguistic variables p and v, which are determined with four initial conditions in (17).

v \ p	A_1	A_2	A_3	A_4	A_5
B_1	C_5	C_5	C_3	C_4	C_1
B_2	C_5	C_5	C_5	C_1	C_1
B_3	C_5	C_5	C_2	C_2	C_1
B_4	C_5	C_4	C_1	C_1	C_1
B_5	C_3	C_3	C_2	C_3	C_1

Figure 7. Optimal control rules of the FLC in Section 4.1, in which membership functions $A_1, A_2, \ldots, A_5$ and $B_1, B_2, \ldots, B_5$ are defined in Figure 6, and $C_1, C_2, \ldots, C_5$ are membership functions of F as defined in Figure 2.

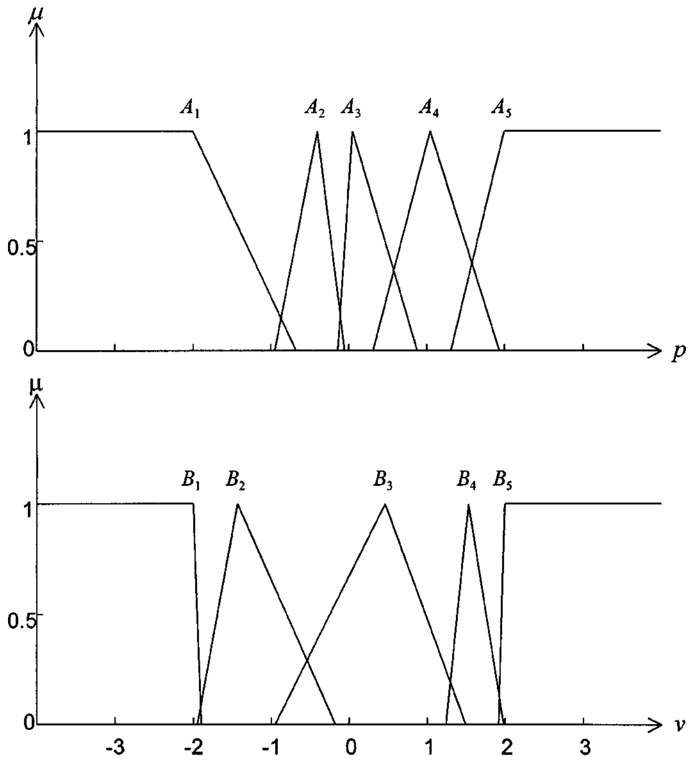


Figure 8. Optimal membership functions of the input linguistic variables p and v, which are determined with sexteen initial conditions in (17).

4.2. THE CASE OF SIXTEEN INITIAL CONDITIONS

To enhance the capability of the designed FLC, more initial points will be included in defining the fitness function. Applying the proposed method to the system with the sixteen initial conditions shown in Figure 4(b), the optimal membership func-

		p				
		A_1	A_2	A_3	A_4	A_5
	B_1	C_4	C_3	C_3	C_4	C_2
	B_2	C_5	C_5	C_4	C_1	C_1
v	B_3	C_5	C_5	C_1	C_1	C_1
	B_4	C_4	C_1	C_1	C_1	C_1
	B_5	C_4	C_1	C_1	C_1	C_1

Figure 9. Optimal control rules of the FLC in Section 4.2, in which membership functions $A_1, A_2, \ldots, A_5$ and $B_1, B_2, \ldots, B_5$ are defined in Figure 8, and $C_1, C_2, \ldots, C_5$ are membership functions of F as defined in Figure 2.

tions and the optimal fuzzy control rules to generate F are determined as shown in Figures 8 and 9, respectively.

4.3. PERFORMANCE OF THE DESIGNED FLC

After obtaining the optimal membership functions and the optimal fuzzy control rules, the performance of the designed FLC should be checked. Since the FLC is designed to determine the force that will drive the cart from an arbitrarily initial condition to the desired final condition, different initial conditions should be chosen for testing. However, it will be hard to choose these test points from the input space properly. From theoretical point of view, all point in the input space should be chosen, but this is impossible to be done in practice. Therefore, a strategy will be adopted in this example, in which a set of points will be chosen uniformly from the input space and then be used for testing. A designed FLC will have better performance if more test points can be driven to the desired final state.

If only a few test points are chosen, then it is very possible that one cannot distinguish the performance between two FLCs. However, too many test points will also lead to very heavy computation. Therefore, for compromise, 1681 (41×41) test points are chosen uniformly from the input space as follows:

$$(p_i, v_j) = \left(\frac{i-1}{10} - 2, \frac{j-1}{10} - 2\right) \quad \text{for } 1 \leqslant i \leqslant 41,\ 1 \leqslant j \leqslant 41. \tag{18}$$

These test points are then put into the designed FLC to check whether they can be driven to the desired final state $(p, v) = (0, 0)$. The test results are shown in Figures 10(a) and (b) for the designed FLCs in Sections 4.1 and 4.2, respectively. The coordinates of Figure 10 are the ith test point, $1 \leqslant i \leqslant 1681$, and its corresponding time steps required to reach the state $(p, v) = (0, 0)$, respectively. As mentioned previously, $200\Delta t$ is the limit of time steps in the simulation. Therefore, the maximal number of times steps required in Figure 10 will be 200, which indicates that the corresponding test point cannot be driven to $(p, v) = (0, 0)$ within $200\Delta t$.

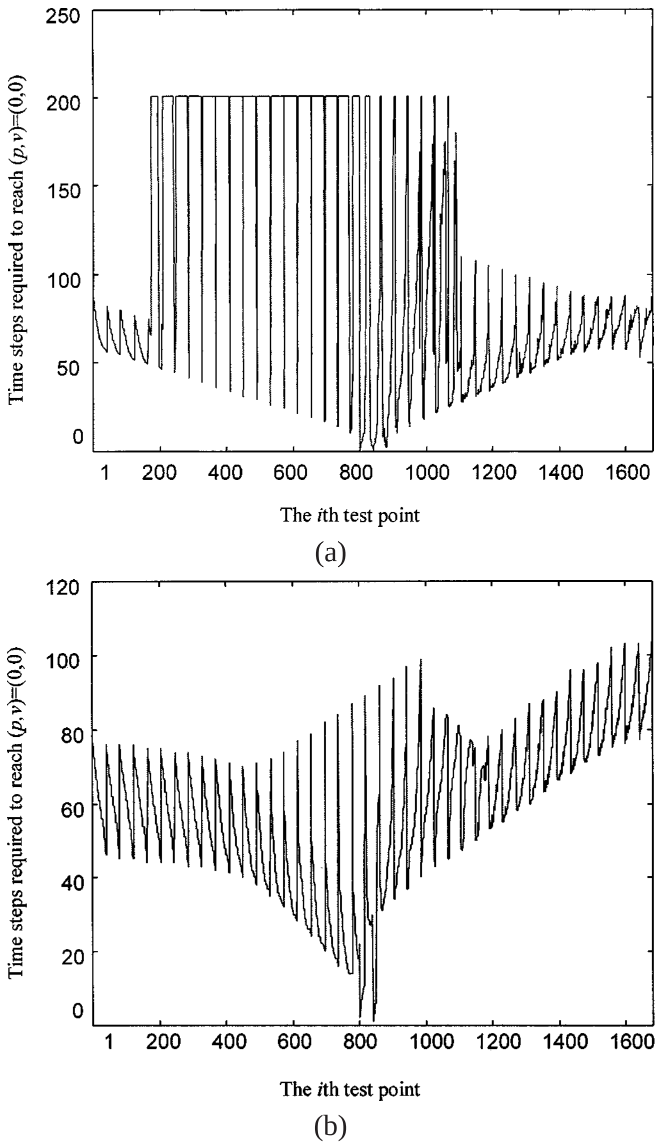


Figure 10. (a) The performance of the FLC in Section 4.1. (b) The performance of the FLC in Section 4.2. The coordinates of these two figures are the ith test point, $1 \leqslant i \leqslant 1681$, and its corresponding time steps required to reach $(p, v) = (0, 0)$. Since $200\Delta t$ is the limit of time steps in the simulation, the maximal number of time steps required will be 200.

From the above illustration and the results in Figures 10(a) and (b), one can find that many test points cannot be driven to the desired final state for the designed FLC in Section 4.1, while the designed FLC in Section 4.2 has no such a problem. Therefore, it is obvious that performance of the FLC in Section 4.2 is better than

Table I. Performance of the designed FLC in Section 4.2 with respect to changing cart mass

Cart mass (kg)	Total time steps	Avg. no. of time steps	Max. no. of time steps	No. of failures	Ratio of success
1	143288	85.24	200	117	93.04%
2	84806	50.45	125	0	100%
3	87597	52.11	129	0	100%
4	104037	61.89	141	0	100%
5	90808	54.02	143	0	100%
6	97750	58.15	147	0	100%
7	98187	58.41	144	0	100%
8	97649	58.09	158	0	100%
9	98406	58.54	145	0	100%
10	95380	56.74	144	0	100%
11	92724	55.16	139	0	100%
12	91362	54.35	167	0	100%
13	92304	54.91	151	0	100%
14	91497	54.43	119	0	100%
15	87395	51.99	95	0	100%
16	89715	53.37	93	0	100%
17	91581	54.48	99	0	100%
18	93615	55.69	100	0	100%
19	95632	56.89	106	0	100%
20	97683	58.11	104	0	100%
21	100070	59.53	114	0	100%
22	102356	60.89	112	0	100%
23	104289	62.04	112	0	100%
24	106609	63.42	119	0	100%
25	109198	64.96	130	0	100%
26	111702	66.45	137	0	100%
27	114005	67.82	138	0	100%
28	116476	69.29	143	0	100%
29	121082	72.03	166	0	100%
30	151324	90.02	180	0	100%
31	183515	109.17	185	0	100%
32	187347	111.45	187	0	100%
33	190995	113.62	194	0	100%
34	195080	116.05	199	0	100%
35	198745	118.23	200	5	99.70%
36	202426	120.42	200	15	99.11%
37	206208	122.67	200	23	98.63%
38	209806	124.81	200	36	97.86%
39	213521	127.02	200	51	96.97%
40	217370	129.31	200	70	95.84%

the one in Section 4.1. This is a result as expected since more initial conditions are used in designing the FLC in Section 4.2.

In addition to the above test, the performance of the designed FLC with respect to changing cart mass will also be checked. The strategy is to use the 1681 test points in (18) to check whether the FLC in Section 4.2 can drive any of the test points to $(p, v) = (0, 0)$ when the cart mass varies within a wide range. Details of the simulation results are shown in Table I, from which one can find that the designed FLC still has a very good performance when the cart mass varies within the range [2, 34 kg] in increments 1 kg.

5. Conclusion and Discussion

A GA-based method is proposed for design of FLCs. The major advantage of this method is that the membership functions and the fuzzy control rules can be determined simultaneously provided that the FLCs take the form of Mamdani implication. In the proposed method, real-coded chromosomes, enlarged sampling space, rank-based selection, convex crossover, and nonuniform mutation are used. For a TISO system that has n triangular membership functions for each of the input linguistic variables, it is shown that a chromosome will contain $(n^2 + 6n - 6)$ genes to be optimized. Moreover, based on a proportional scaling method, it is also shown that the range of each gene can be chosen almost arbitrarily provided that the lower bound is greater than zero.

In the simulation example, the proposed method is applied to solve the cart-centering problem. The results show that if only 4 initial conditions are chosen in defining the fitness function, then the designed FLC will have difficulty in driving the system from any one of the 1681 test points to the desired final one. However, if 16 different initial conditions are taken into account in defining the fitness function, then the designed FLC will be able to drive the system to the desired final point from any one of the 1681 test points. The comparison between these two cases shows that the designed FLC will have better performance if more initial conditions are included in defining the fitness function. Moreover, the performance of the designed FLC with respect to changing cart mass is also checked. The simulation results show that the designed FLC works well even when the cart mass varies within a wide range.

In defining the membership functions for the output linguistic variable, for simplification of the defuzzification process, these functions are assumed to be symmetric and evenly distributed. However, the left widths, the right widths, and the locations of the peaks of these functions can also be chosen as parameters to be optimized if one is interested in the study of this issue. The only disadvantage that may arise will be the increase in complexity of the problem. Meanwhile, for the cart-centering problem, how to determine the number and the proper set of initial conditions in defining the fitness function will be another issue for further study.

Acknowledgements

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grants NSC88-2213-E-224-035 and NSC88-2213-E-224-037.

References

1. Chang, C. H. and Wu, Y. C.: The genetic algorithm-based tuning method for symmetric membership functions of fuzzy logic control systems, in: *Proc. of the Internat. IEEE/IAS Conf. on Industrial Automation and Control: Emerging Technologies*, 1995, pp. 421-428.
2. Fogel, D.: An introduction to simulated evolutionary optimization, *IEEE Trans. Neural Networks* **5** (1994), 3–14.
3. Gen, M. and Cheng, R.: *Genetic Algorithm and Engineering Design*, Wiley, New York, 1997.
4. Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
5. Grefenstette, J. J.: Optimization of control parameters for genetic algorithms, *IEEE Trans. Systems Man Cybernet.* **16**(1) (1986), 122–128.
6. Holland, J.: *Adaptation in Natural and Artificial System*, Univ. of Michigan Press, Ann Arbor, 1975.
7. Homaifar, A. and McCormick, E.: Simultaneous design of membership functions and rule sets for fuzzy controller using genetic algorithms, *IEEE Trans. Fuzzy Systems* **3**(2) (1995), 129–139.
8. Karr, C. L. and Gentry, E. J.: Fuzzy control of pH using genetic algorithms, *IEEE Trans. Fuzzy Systems* **1**(1) (1993), 46–53.
9. Lee, C. C.: Fuzzy logic in control systems: Fuzzy logic controller – Part I, *IEEE Trans. Systems Man Cybernet.* **20** (1990), 404–418.
10. Mamdani, E. H.: Applications of fuzzy algorithms for control of simple dynamic plant, *Proc. IEE* **121**(12) (1974), 1585–1588.
11. Man, K. F., Tang, K. S., and Kwong, S.: Genetic algorithms: Concepts and applications, *IEEE Trans. Industr. Electronics* **43**(5) (1996), 519–534.
12. Michalewicz, Z.: *Genetic Algorithm + Data Structure = Evolution Programs*, 2nd ed., Springer, New York, 1994.
13. Schwefel, H.: *Evolution and Optimum Seeking*, Wiley, New York, 1994.
14. Tang, K. S., Chan, C. Y., and Man, K. F.: A simultaneous method for fuzzy memberships and rules optimization, in: *Proc. of the IEEE Internat. Conf. on Industrial Technology*, 1996, pp. 279–283.
15. Thrift, P.: Fuzzy logic synthesis with genetic algorithms, in: *Proc. of the 4th Internat. Conf. on Genetic Algorithms*, 1991, pp. 450–457.
16. Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Systems Man Cybernet.* **3** (1973), 28–44.