PSO-based Motion Fuzzy Controller Design for Mobile Robots

Ching-Chang Wong, Hou-Yi Wang, and Shih-An Li

Abstract

A motion control structure with a distance fuzzy controller and an angle fuzzy controller is proposed to determine velocities of the left-wheeled motor and right-wheeled motor of the two-wheeled mobile robot. A PSO-based method is proposed to automatically determine appropriate membership functions of these two fuzzy systems so that the controlled robot can move to any desired position effectively in a two-dimensional space. A ratio coefficient coding method and a variable fitness function are considered in the proposed PSO-based fuzzy controller design method so that the proposed PSO-based method with a good searching efficiency and the selected fuzzy controller with a good control performance.

Keywords: Particle Swarm Optimization (PSO) Algorithm, Fitness Function, Fuzzy Control, Mobile Robot.

1. Introduction

Many motion controller design methods are proposed for two-wheeled robots so that they can move efficiently in a two-dimensional space [1-8]. One motion control problem of two-wheeled mobile robots is how to independently control the left-wheeled motor and right-wheeled motor. Many controller design methods are based on the fuzzy system concepts and the evolutionary computation technologies. Fuzzy systems can be applied in many fields. Fuzzy controllers have been proven to be good tools for real-time industrial processes, where they are difficult to obtain mathematical models. Genetic algorithm (GA) is one popular evolutionary computation technique to obtain an optimal solution. GA-based fuzzy controller design approaches have been proposed and applied in many different fields [9-12]. Particle Swarm Optimization (PSO) algorithm proposed by Kennedy and Eberhart [13] is another kind of evolutionary computation techniques. The motivation for the development of this method was based on the simulation of simplified animal social behaviors such as fish schooling and bird flocking. In this paper, a fuzzy control structure and a PSO-based method is proposed to determine the velocities of the left-wheeled motor and right-wheeled motor. Similar to Genetic Algorithms, the PSO algorithm is initial with a population of particles which are randomly generated in a search space. However in the PSO algorithm, there is no direct recombination of genetic material between particles during the searching process. The PSO algorithm works on a social behavior of particles in the swarm. It finds the global best solution by simply adjusting the trajectory of each particle toward its own best particle and toward the best particle of the entire swarm at each generation. The PSO algorithm is becoming very popular due to it can be easily implemented and quickly find a good solution [14-18].

In this paper, a PSO-based motion fuzzy controller design method is proposed to automatically determine appropriate membership functions of fuzzy systems to control a two-wheeled mobile robot so that it move efficiently in a two-dimensional space. The rest of this paper is organized as follows: In Section 2, a two-wheeled mobile robot is described and a motion fuzzy control structure is proposed to determine velocities of its left-wheeled motor and right-wheeled motor. In the motion fuzzy control structure, two 2-input-and-1-output fuzzy systems: a distance fuzzy controller and an angle fuzzy controller are used to reduce the design complexity. In Section 3, a PSO-based fuzzy controller design method with a ratio coefficient coding method and a variable fitness function is proposed to automatically select the input and output membership functions of these two fuzzy systems. In Section 4, some results simulated in the 3D robot soccer simulator of FIRA [19] are presented to illustrate the efficiency of the proposed method. Finally, some conclusions are made in Section 5.

2. Motion Fuzzy Control Structure

In this paper, the control system block diagram for a two-wheeled mobile robot can be described in Figure 1. A schematic diagram of the considered two-wheeled mobile robot is described in Figure 2, where x_w-y_w is the world coordinates and x_m-y_m is the local coordinates which is fixed to the robot with its center p as the origin. Its body is symmetric and the center of mass is at the

Corresponding Author: Ching-Chang Wong is with the Department of Electrical Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan.
E-mail: wong@ee.tku.edu.tw
Manuscript received 31 Jan. 2008.

geometric center p of the body. R is the radius of the wheel and L is the displacement from the center of robot to the center of wheel. The set (x, y) represents the position of the geometric center p in the world x_w-y_w coordinates, and the angle θ indicates the orientation of the robot. The angle θ is taken counterclockwise from the x_w-axis to the x_m-axis.

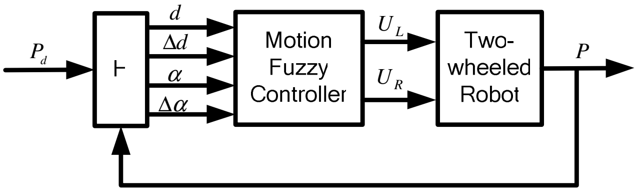


Figure 1. Control system block diagram of the two-wheeled mobile robot.

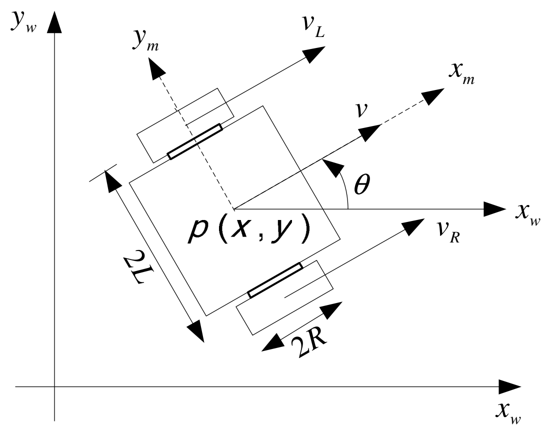


Figure 2. Description of a two-wheeled robot for the motion control.

The robot has two degrees of freedom in its relative position and orientation, so the robot posture can be represented by the vector (x, y, θ). Two postures of the robot are described in Figure 3, where $p(x, y, \theta)$ and $p_d(x_d, y_d, \theta_d)$ denote the current posture and the desired (target) posture of the robot, respectively. θ_d is the angle of robot at the desired position. d is the distance between the current position (x, y) and the desired position (x_d, y_d) of the robot, θ is the angle between the x_w-axis and the head direction of the robot, φ is the angle between the x_w-axis and the direction of the current position of the robot toward the desired position of the robot, and α is the angle between the head direction of the robot and the direction of the current position of the robot toward the desired position of the robot. The following equations are considered so that $\alpha \in [-180,180]$ in this proposed structure.

$$d = \sqrt{(x_d - x)^2 + (y_d - y)^2} \tag{1}$$

$$\alpha = \begin{cases} \varphi - \theta \quad , & if \quad \varphi - 180 < \theta < 180 \\ (\varphi - \theta) - 360, & if \quad -180 < \theta < \varphi - 180 \end{cases} \tag{2}$$

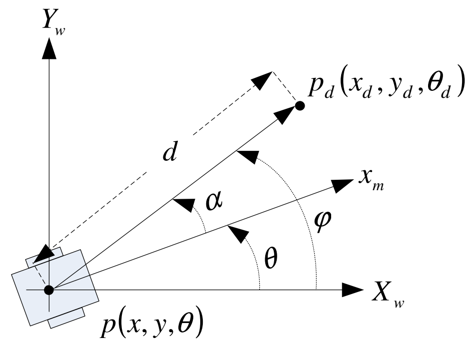


Figure 3. Description of the current posture $p(x, y, \theta)$ and the desired posture $p_d(x_d, y_d, \theta_d)$ of the robot.

Two wheels are fixed in the considered mobile robot and each wheel is independently controlled by each motor, so the mobile path of the controlled robot is determined by the velocities and the directions of these two motors. In this paper, the design object is to determine the velocities U_L and U_R of the left-wheeled motor and right-wheeled motor so that the two-wheeled robot can effectively move from the current position (x, y) to the desired position (x_d, y_d). As shown in Figure 1, the 4-input-and-2-output motion fuzzy controller with four input variables: d (the distance between the robot and the destination), Δd (the change quantity of motion distance), α (the rotation angle of robot), and $\Delta\alpha$ (the change quantity of rotation angle) are considered to determine two velocities U_L and U_R to control the left-wheeled motor and right-wheeled motor, respectively. When the values d and α are respectively determined by Equations (1) and (2), then the values $\Delta\alpha$ and Δd can be respectively calculated by

$$\Delta\alpha(t) = \alpha(t) - \alpha(t-1) \tag{3}$$

and

$$\Delta d(t) = d(t) - d(t-1) \tag{4}$$

Let d, Δd α, and $\Delta\alpha$, are respectively represented by $x_1, x_2, x_3,$ and x_4. As shown in Figure 4, in order to reduce the design complexity, the 4-input-and-2-output fuzzy controller as shown in Figure 1 is decomposed into two 2-input-and-1-output fuzzy controllers named distance fuzzy controller and angle fuzzy controller. In the distance fuzzy controller, x_1 (d) and x_2 (Δd) are used to be the input variables and y_1 is the output variable of this fuzzy system. In the angle fuzzy controller, x_3 (α) and x_4 ($\Delta\alpha$) are used to be the input variables and y_2 is the output variable of this fuzzy system. When y_1 and y_2 are obtained by these two fuzzy controllers, U_L

and U_R can be obtained by

$$U_L = y_1 + y_2 \tag{5}$$

and

$$U_R = y_1 - y_2 \tag{6}$$

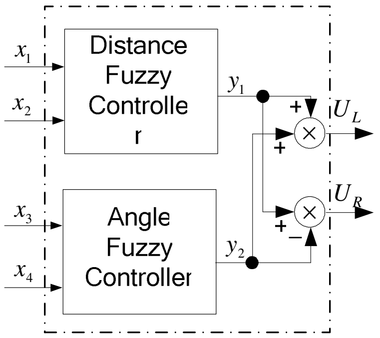


Figure 4. System block of the motion fuzzy controller.

The fuzzy rule bases of the distance fuzzy controller and the angle fuzzy controller are respectively described in Table 1 and Table 2. The fuzzy rules of the distance fuzzy controller and the angle fuzzy controller can be respectively described by

$$R_{y_1}(j_1, j_2): \text{IF } x_1 \text{ is } A_{(1,j_1)} \text{ AND } x_2 \text{ is } A_{(2,j_2)}, \text{THEN } y_1 \text{ is } y_{1(j_1,j_2)} \quad j_1, j_2 \in \{-3,-2,-1,0,1,2,3\} \tag{7}$$

$$R_{y_2}(j_3, j_4): \text{IF } x_3 \text{ is } A_{(3,j_3)} \text{ AND } x_4 \text{ is } A_{(4,j_4)}, \text{THEN } y_2 \text{ is } y_{2(j_3,j_4)} \quad j_3, j_4 \in \{-3,-2,-1,0,1,2,3\} \tag{8}$$

where $x_1, x_2, x_3,$ and x_4 are input variables, y_1 and y_2 are output variables, $A_{(1,j_1)} \in T(x_1)$, $A_{(2,j_2)} \in T(x_2)$, $y_{1(j_1,j_2)} \in T(y_1)$, $A_{(3,j_3)} \in T(x_3)$, $A_{(4,j_4)} \in T(x_4)$, and $y_{2(j_1,j_2)} \in T(y_2)$. The following term sets are used to describe the fuzzy sets of each input and output fuzzy variables:

$$T(x_i) = \{NB, NM, NS, ZO, PS, PM, PB\}, i = 1,2,3,4 \\ = \{A_{(i,-3)}, A_{(i,-2)}, A_{(i,-1)}, A_{(i,0)}, A_{(i,1)}, A_{(i,2)}, A_{(i,3)}\} \tag{9}$$

$$T(y_m) = \{NB, NM, NS, ZO, PS, PM, PB\}, m = 1,2 \\ = \{y_{(m,-3)}, y_{(m,-2)}, y_{(m,-1)}, y_{(m,0)}, y_{(m,1)}, y_{(m,2)}, y_{(m,3)}\} \tag{10}$$

That is every variables have seven linguistic values: Negative Big (NB), Negative Middle (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Middle (PM), and Positive Big (PB). As shown in Figure 5, the triangle membership functions and the singleton membership functions are used to describe the fuzzy sets of input variables and the output variables, respectively. Based on the weighted average method, the final outputs of the distance fuzzy controller and angle fuzzy controller can be respectively described by:

$$y_1 = \sum_{j_1=-3}^{3} \sum_{j_2=-3}^{3} w_{(j_1,j_2)} y_{1(j_1,j_2)} \tag{11}$$

and

$$y_2 = \sum_{j_3=-3}^{3} \sum_{j_4=-3}^{3} w_{(j_3,j_4)} y_{1(j_3,j_4)} \tag{12}$$

where $w_{(j_1,j_2)}$ and $w_{(j_3,j_4)}$ are respectively described by

$$w_{(j_1,j_2)} = \frac{\min(\mu_{A_{(1,j_1)}}(x_1), \mu_{A_{(2,j_2)}}(x_2))}{\sum_{j_1=-3}^{3} \sum_{j_2=-3}^{3} \min(\mu_{A_{(1,j_1)}}(x_1), \mu_{A_{(2,j_2)}}(x_2))} \tag{13}$$

and

$$w_{(j_3,j_4)} = \frac{\min(\mu_{A_{(3,j_3)}}(x_3), \mu_{A_{(4,j_4)}}(x_4))}{\sum_{j_3=-3}^{3} \sum_{j_4=-3}^{3} \min(\mu_{A_{(3,j_3)}}(x_3), \mu_{A_{(4,j_4)}}(x_4))} \tag{14}$$

Table 1. Fuzzy rule base of distance fuzzy controller.

y_1		x_1						
		NB	NM	NS	ZO	PS	PM	PB
x_2	PB	ZO $y_{1(-3,3)}$	PS $y_{1(-2,3)}$	PM $y_{1(-1,3)}$	PB $y_{1(0,3)}$	PB $y_{1(1,3)}$	PB $y_{1(2,3)}$	PB $y_{1(3,3)}$
	PM	NS $y_{1(-3,2)}$	ZO $y_{1(-2,2)}$	PS $y_{1(-1,2)}$	PM $y_{1(0,2)}$	PB $y_{1(1,2)}$	PB $y_{1(2,2)}$	PB $y_{1(3,2)}$
	PS	NS $y_{1(-3,1)}$	NS $y_{1(-2,1)}$	ZO $y_{1(-1,1)}$	PS $y_{1(0,1)}$	PM $y_{1(1,1)}$	PB $y_{1(2,1)}$	PB $y_{1(3,1)}$
	ZO	NB $y_{1(-3,0)}$	NM $y_{1(-2,0)}$	NS $y_{1(-1,0)}$	ZO $Y_{1(0,0)}$	PS $y_{1(1,0)}$	PM $y_{1(2,0)}$	PB $y_{1(3,0)}$
	NS	NB $y_{1(-3,-1)}$	NB $y_{1(-2,1)}$	NM $y_{1(-1,-1)}$	NS $y_{1(0,-1)}$	ZO $y_{1(1,-1)}$	PS $y_{1(2,-1)}$	PM $y_{1(3,-1)}$
	NM	NB $y_{1(-3,-2)}$	NB $y_{1(-2,2)}$	NB $y_{1(-1,-2)}$	NM $y_{1(0,-2)}$	NS $y_{1(1,-2)}$	ZO $y_{1(2,-2)}$	PS $y_{1(3,-2)}$
	NB	NB $y_{1(-3,-3)}$	NB $y_{1(-2,3)}$	NB $y_{1(-1,-3)}$	NB $y_{1(0,-3)}$	NM $y_{1(1,-3)}$	NS $y_{1(2,-3)}$	ZO $y_{1(3,-3)}$

Table 2. Fuzzy rule base of angle fuzzy controller.

y_2		x_3						
		NB	NM	NS	ZO	PS	PM	PB
x_4	PB	ZO $y_{2(-3,3)}$	PS $y_{2(-2,3)}$	PM $y_{2(-1,3)}$	PB $y_{2(0,3)}$	PB $y_{2(1,3)}$	PB $y_{2(2,3)}$	PB $y_{2(3,3)}$
	PM	NS $y_{2(-3,2)}$	ZO $y_{2(-2,2)}$	PS $y_{2(-1,2)}$	PM $y_{2(0,2)}$	PB $y_{2(1,2)}$	PB $y_{2(2,2)}$	PB $y_{2(3,2)}$
	PS	NS $y_{2(-3,1)}$	NS $y_{2(-2,1)}$	ZO $y_{2(-1,1)}$	PS $y_{2(0,1)}$	PM $y_{2(1,1)}$	PB $y_{2(2,1)}$	PB $y_{2(3,1)}$
	ZO	NB $y_{2(-3,0)}$	NM $y_{2(-2,0)}$	NS $y_{2(-1,0)}$	ZO $y_{2(0,0)}$	PS $y_{2(1,0)}$	PM $y_{2(2,0)}$	PB $y_{2(3,0)}$
	NS	NB $y_{2(-3,-1)}$	NB $y_{2(-2,1)}$	NM $y_{2(-1,-1)}$	NS $y_{2(0,-1)}$	ZO $y_{2(1,-1)}$	PS $y_{2(2,-1)}$	PM $y_{2(3,-1)}$
	NM	NB $y_{2(-3,-2)}$	NB $y_{2(-2,2)}$	NB $y_{2(-1,-2)}$	NM $y_{2(0,-2)}$	NS $y_{2(1,-2)}$	ZO $y_{2(2,-2)}$	PS $y_{2(3,-2)}$
	NB	NB $y_{2(-3,-3)}$	NB $y_{2(-2,3)}$	NB $y_{2(-1,-3)}$	NB $y_{2(0,-3)}$	NM $y_{2(1,-3)}$	NS $y_{2(2,-3)}$	ZO $y_{2(3,-3)}$

When the input data of $x_1, x_2, x_3,$ and x_4 are given, y_1 and y_2 can be determined by Equations (11) and (12), respectively. Therefore, the left-wheel velocity U_L and the right-wheel velocity U_R can be obtained by

Equations (5) and (6), respectively. The next posture $(x(k+1), y(k+1), \theta(k+1))$ of the mobile robot can be determined by the 3D robot soccer simulator of FIRA.

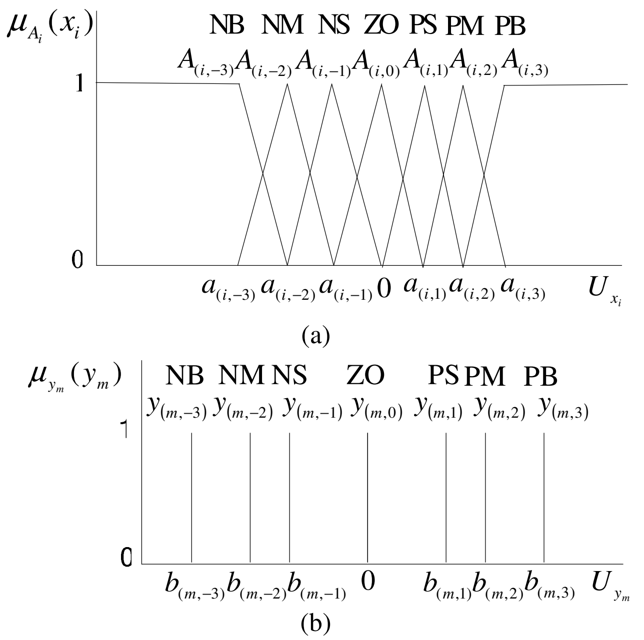


Figure 5. Membership functions: (a) the antecedent fuzzy sets for x_i, (b) the consequent fuzzy sets for y_m.

3. PSO-based Fuzzy Controller Design Method

The PSO algorithm is a computation technique proposed by Kennedy and Eberhart. Its development was based on observations of the social behavior of animals such as bird flocking and fish schooling of the swarm theory. If a set p^g with N particles is called a population in the g-th generation and expressed by

$$p^g = \{p_1^g, p_2^g, \ldots, p_h^g, \ldots, p_N^g\} \quad (15)$$

The position vector and velocity vector of the h-th particle ($h \in \{1,2\ldots,N\}$) in the g-th generation ($g \in \{1,2,\ldots,G\}$) are respectively denoted by

$$p_h^g = (p_{(h,1)}^g, p_{(h,2)}^g, \ldots, p_{(h,j)}^g, \ldots, p_{(h,n)}^g) \quad (16)$$

and

$$v_h^g = (v_{(h,1)}^g, v_{(h,2)}^g, \ldots, v_{(h,j)}^g, \ldots, v_{(h,n)}^g) \quad (17)$$

where n is the number of searching parameters and $p_{(h,j)}^g$ denotes the position of the j-th parameter ($j \in \{1,2,\ldots n\}$) of the h-th particle in the g-th generation. The procedure of PSO algorithm can be described as follows:

Step 1: Initialize the PSO algorithm by setting g=1, $F_1^{pbest} = F_2^{pbest} = \cdots = F_N^{pbest} = 0$, the maximum number of generation (G), the number of particles (N), and four parameter values of c_1, c_2, $\omega_{\max}$, and $\omega_{\min}$.

Step 2: Generate the initial position vector $p_h^1 = (p_{(h,1)}^1, p_{(h,2)}^1, \ldots, p_{(h,j)}^1, \ldots, p_{(h,n)}^1)$ and the initial velocity vector $v_h^1 = (v_{(h,1)}^1, v_{(h,2)}^1, \ldots, v_{(h,j)}^1, \ldots, v_{(h,n)}^1)$ of N particles randomly by

$$p_{(h,j)}^1 = p_j^{\min} + (p_j^{\max} - p_j^{\min})rand() \quad (18)$$

and

$$v_{(h,j)}^1 = \frac{(v_j^{\max} - v_j^{\min})}{20} rand() \quad (19)$$

where $p_j^{\max}$ and $p_j^{\min}$ are respectively the maximum value and minimum value of the j-th parameter (i.e., $p_j \in [p_j^{\min}, p_j^{\max}]$). $v_j^{\max}$ and $v_j^{\min}$ are the maximum velocity and minimum velocity the j-th parameter. *rand*() is an uniformly distributed random number in [0,1].

Step 3: Calculate the fitness value of each particle in the g-th generation by

$$F(p_h^g) = fit(p_h^g), h = 1,2,\ldots,N \quad (20)$$

where *fit*(.) is the fitness function.

Step 4: Determine F_h^{Pbest} and p_h^{Pbest} for each particle by

$$F_h^{Pbest} = \begin{cases} F_h^g, & \text{if } F_h^{Pbest} \le F_h^g \\ F_h^{Pbest}, & \text{otherwise} \end{cases}, h \in \{1,2,\ldots,N\} \quad (21)$$

and

$$p_h^{Pbest} = \begin{cases} p_h^g, & \text{if } F_h^{Pbest} \le F_h^g \\ p_h^{Pbest}, & \text{otherwise} \end{cases}, h \in \{1,2,\ldots,N\} \quad (22)$$

where p_h^{Pbest} is the position vector of the h-th particle with the personal best fitness value F_h^{Pbest} from the beginning to the current generation.

Step 5: Find an index q of the particle with the highest fitness by

$$q = \arg \max_{h \in \{1,2,\ldots,N\}} F_h^{Pbest} \quad (23)$$

and determine F^{Gbest} and P^{Gbest} by

$$F^{Gbest} = F_q^{Pbest} = \max_{h \in \{1,2,\ldots,N\}} F_h^{Pbest} \quad (24)$$

and

$$p^{Gbest} = p_q^{Pbest} \quad (25)$$

where p^{Gbest} is the position vector of the particle with the global best fitness value F^{Gbest} from the beginning to the current generation.

Step 6: If g=G, then go to Step 12, Otherwise, go to Step

7.

Step 7: Update the velocity vector of each particle by

$$v_h^{g+1} = \omega \cdot v_h^g + c_1 \cdot rand1() \cdot (p^{Gbest} - p_h^g) + c_2 \cdot rand2() \cdot (p_h^{Pbest} - p_h^g) \quad (26)$$

where v_h^g and is the current velocity vector of the h-th particle in the g-th generation. v_h^{g+1} is the next velocity vector of the h-th particle in the (g+1)-th generation. *rand1*() and *rand2*() are two uniformly distributed random numbers in [0,1]. c_1 and c_2 are constant values. ω is a weight value and defined by

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{G} \cdot g \quad (27)$$

where $\omega_{\max}$ and $\omega_{\min}$ are respectively a maximum value and a minimum value of ω.

Step 8: Check the velocity constraint by

$$v_{(h,j)}^{g+1} = \begin{cases} v_j^{\max}, & \text{if } v_{(h,j)}^{g+1} > v_j^{\max} \\ v_{(h,j)}^{g+1}, & \text{if } v_j^{\min} \le v_{(h,j)}^{g+1} \le v_j^{\max} \\ v_j^{\min}, & \text{if } v_{(h,j)}^{g+1} < v_j^{\min} \end{cases} \quad (28)$$

$$h = 1,2,...,N, \quad j = 1,2,...,n$$

Step 9: Update the position vector of each particle by

$$p_h^{g+1} = p_h^g + v_h^{g+1} \quad (29)$$

where p_h^g is the current position vector of the h-th particle in the g-th generation. p_h^{g+1} is the next position vector of the h-th particle in the (g+1)-th generation.

Step10: Bound the updated position vector of each particle in the searching range by

$$p_{(h,j)}^{g+1} = \begin{cases} p_j^{\max}, & \text{if } p_{(h,j)}^{g+1} > p_j^{\max} \\ p_{(h,j)}^{g+1}, & \text{if } p_j^{\min} \le p_{(h,j)}^{g+1} \le p_j^{\max} \\ p_j^{\min}, & \text{if } p_{(h,j)}^{g+1} < p_j^{\min} \end{cases} \quad (30)$$

$$h = 1,2,...,N, \quad j = 1,2,...,n$$

Step11: Let g=g+1 and go to Step 3.

Step12: Determine the corresponding fuzzy controller based on the position of the particle p^{Gbest} with the best fitness value F^{Gbest}.

How to appropriately code the searching parameters and design a fitness function are two important key points for the PSO algorithm to effectively search a good result. In this paper, a ratio coefficient coding method and a variable fitness function are proposed so that input and output membership functions of the fuzzy controller can be effectively selected. They are described as follows:

In the proposed motion fuzzy control structure described in Section 2, seven linguistic items are considered for the input and output variables as described in Figures 5. If these membership functions have the symmetric properties. For example in Figure 5, $a_{(i,-3)} = -a_{(i,3)}$, $a_{(i,-2)} = -a_{(i,2)}$, $a_{(i,-1)} = -a_{(i,1)}$ and only three values are needed to determine all the membership functions for each variable. Thus, three sets of three center values of each membership functions $\{a_{(1,1)}, a_{(1,2)}, a_{(1,3)}\}$, $\{a_{(2,1)}, a_{(2,2)}, a_{(2,3)}\}$, and $\{b_{(1,1)}, b_{(1,2)}, b_{(1,3)}\}$, are respectively used to determine input and output membership functions of $x_1, x_2,$ and y_1 for a distance fuzzy controller described in Equation (7). Similarly, $\{a_{(3,1)}, a_{(3,2)}, a_{(3,3)}\}$, $\{a_{(4,1)}, a_{(4,2)}, a_{(4,3)}\}$, and $\{b_{(2,1)}, b_{(2,2)}, b_{(2,3)}\}$, are respectively used to determine input and output membership functions of $x_3, x_4,$ and y_2 for an angle fuzzy controller described in Equation (8). In this paper, the following ratio coefficient coding method is proposed to determine these center values of $a_{(1,j_1)}$, $a_{(2,j_2)}$, and $b_{(1,j_1)}$ so that all the membership functions of these three variables $x_1, x_2,$ and y_1 are determined.

$$a_{(1,j_1)} = \begin{cases} -x_1^{\max} \prod_{p=-j_1}^{3} k_{(1,p)} & if \quad -3 \le j_1 < 0 \\ 0 & if \quad j_1 = 0 \\ x_1^{\max} \prod_{p=j_1}^{3} k_{(1,p)} & if \quad 0 < j_1 \le 3 \end{cases}, \quad (31)$$

$$a_{(2,j_2)} = \begin{cases} -x_2^{\max} \prod_{p=-j_2}^{3} k_{(2,p)} & if \quad -3 \le j_2 < 0 \\ 0 & if \quad j_2 = 0 \\ x_2^{\max} \prod_{p=j_2}^{3} k_{(2,p)} & if \quad 0 < j_2 \le 3 \end{cases}, \quad (32)$$

and

$$b_{(1,j_1)} = \begin{cases} -y_1^{\max} \prod_{p=-j_1}^{3} k_{(5,p)} & if \quad -3 \le j_1 < 0 \\ 0 & if \quad j_1 = 0 \\ y_1^{\max} \prod_{p=j_1}^{3} k_{(5,p)} & if \quad 0 < j_1 \le 3 \end{cases}, \quad (33)$$

where $k_{(i,j)} \in [0,1], i = 1,2,5, j = 1,2,3,$ are ratio coefficients, $x_1 \in [-x_1^{\min}, x_1^{\max}]$, $x_2 \in [-x_2^{\min}, x_2^{\max}]$, and $y_1 \in [-y_1^{\min}, y_1^{\max}]$. That is, three ratio coefficients are needed to decide all the membership functions for each variable so that only nine ratio coefficient values of $k_{(1,1)}, k_{(1,2)}, k_{(1,3)}, k_{(2,1)}, k_{(2,2)}, k_{(2,3)}, k_{(5,1)}, k_{(5,2)},$ and $k_{(5,3)}$ are needed to determine all the input and output membership functions of $x_1, x_2,$ and y_1 for a distance fuzzy controller. Similarly based on the ratio coefficient coding method, nine ratio coefficient values of $k_{(3,1)}, k_{(3,2)}, k_{(3,3)}, k_{(4,1)}, k_{(4,2)}, k_{(4,3)}, k_{(6,1)}, k_{(6,2)}$, and $k_{(6,3)}$ described in the following three equations are used to determine these center values of $a_{(3,j_3)}$, $a_{(4,j_4)}$, and $b_{(2,j_2)}$ so that all the input and output membership functions of x_3, x_4, and y_2 for an angle fuzzy controller are determined.

$$a_{(3,j_3)} = \begin{cases} -x_1^{\max}\prod_{p=-j_3}^{3} k_{(3,p)} & if \quad -3 \le j_3 < 0 \\ 0 & if \quad j_3 = 0 \\ x_1^{\max}\prod_{p=j_3}^{3} k_{(3,p)} & if \quad 0 < j_3 \le 3 \end{cases}, \quad (34)$$

$$a_{(4,j_4)} = \begin{cases} -x_2^{\max}\prod_{p=-j_5}^{3} k_{(4,p)} & if \quad -3 \le j_4 < 0 \\ 0 & if \quad j_4 = 0 \\ x_2^{\max}\prod_{p=j_4}^{3} k_{(4,p)} & if \quad 0 < j_4 \le 3 \end{cases}, \quad (35)$$

and

$$b_{(2,j_2)} = \begin{cases} -y_1^{\max}\prod_{p=-j_2}^{3} k_{(6,p)} & if \quad -3 \le j_2 < 0 \\ 0 & if \quad j_2 = 0 \\ y_1^{\max}\prod_{p=j_2}^{3} k_{(6,p)} & if \quad 0 < j_2 \le 3 \end{cases}, \quad (36)$$

The position vector of each particle in the PSO algorithm represents a fuzzy system and each fuzzy system has three linguistic variables, thus the position vector of the *h*-th particle in the *g*-th generation described in Equation (16) for a distance fuzzy system and an angle fuzzy system can be respectively expressed by

$$p_h^g = \{p_{(h,1)}^g, p_{(h,2)}^g, \ldots, p_{(h,9)}^g\} = \{k_{(h,1,1)}^g, k_{(h,1,2)}^g, k_{(h,1,3)}^g, k_{(h,2,1)}^g, k_{(h,2,2)}^g, k_{(h,2,3)}^g, k_{(h,5,1)}^g, k_{(h,5,2)}^g, k_{(h,5,3)}^g\} \quad (37)$$

and

$$p_h^g = \{p_{(h,1)}^g, p_{(h,2)}^g, \ldots, p_{(h,9)}^g\} = \{k_{(h,3,1)}^g, k_{(h,3,2)}^g, k_{(h,3,3)}^g, k_{(h,4,1)}^g, k_{(h,4,2)}^g, k_{(h,4,3)}^g, k_{(h,6,1)}^g, k_{(h,6,2)}^g, k_{(h,6,3)}^g\} \quad (38)$$

where $k_{(h,i,j)}^g \in [0,1]$ and there are nine parameters (n=9) needed to be selected by the PSO algorithm for each fuzzy system. That is, $\{k_{(h,1,1)}^g, k_{(h,1,2)}^g, k_{(h,1,3)}^g\}$, $\{k_{(h,2,1)}^g, k_{(h,2,2)}^g, k_{(h,2,3)}^g\}$, and $\{k_{(h,5,1)}^g, k_{(h,5,2)}^g, k_{(h,5,3)}^g\}$ in Equation (37) are respectively used to determine input and output membership functions of x_1, x_2, and y_1 for a distance fuzzy system. Similarly, $\{k_{(h,3,1)}^g, k_{(h,3,2)}^g, k_{(h,3,3)}^g\}$, $\{k_{(h,4,1)}^g, k_{(h,4,2)}^g, k_{(h,4,3)}^g\}$, and $\{k_{(h,6,1)}^g, k_{(h,6,2)}^g, k_{(h,6,3)}^g\}$ in Equation (38) are respectively used to determine input and output membership functions of x_3, x_4, and y_2 for an angle fuzzy system.

Moreover, a variable fitness function in the proposed PSO-based motion fuzzy controller design method is proposed and described by

$$fit(P_h^g) = \exp\left(-\left(\frac{Tr(P_h^g)}{\delta_1^g}\right)^2\right)\cdot\exp\left(-\left(\frac{IAE(P_h^g)}{\delta_2^g}\right)^2\right) \quad (39)$$

where *Tr*(p_h^g) and *IAE*(p_h^g) denote the rise time (*Tr*) and the integral absolute error (*IAE*) of the control performance of the fuzzy controller corresponding to the position vector p_h^g, respectively. The fitness function described by Equation (39) is variable because δ_1^g and δ_2^g are two variables which are respectively determined by

$$\delta_1^g = Tr(P^{Gbest}) \quad (40)$$

and

$$\delta_2^g = IAE(P^{Gbest}) \quad (41)$$

where P^{Gbest} is the particle with the global best fitness value.

4. Simulation Results

In the parameter setting of the PSO algorithm, we let G=50, N=40, $c_1 = c_2 = 2$, $\omega_{\max} = 0.9$, and $\omega_{\min} = 0.2$. Two motion control methods designed by GA and PSO are compared in this section, where the membership functions of two fuzzy systems in the proposed structure are design by the GA-based method and the PSO-based method, respectively. They are simulated in the 3D robot soccer simulator of FIRA and the control results are analysis by the MATLAB. Membership functions of d, Δd, U_d, α, $\Delta\alpha$, and U_α determined by the proposed PSO-based method are described in Figure 6 and Figure 7, the results of the best ratio coefficient of the distance fuzzy controller is (0.53, 0.51, 0.49, 0.39, 0.40, 0.75, 0.73, 0.86, 0.88) and the angle fuzzy controller is (0.81, 0.67, 0.71, 0.52, 0.41, 0.56, 0.29, 0.59, 0.46) respectively, Four trajectories of robot controlled by the proposed motion fuzzy controller selected by the PSO-based method are shown in Figure 8, where the desired position is (0, 0) and four initial postures $p(x, y, \theta)$ of the robot are (-65, 60, 180), (55, 60, 180), (-55, -90, 180), and (65, -30, 180). Consider the initial posture (55, 60, 180), the control performance of the controller selected by the GA-based method and PSO-based method in the angle and distance control are described in Table 3 and Table 4, respectively. Their search and control results in the angle and distance control are shown in Figure 9 and Figure 10, respectively. We can see that the PSO-based method has the ability to quickly converge to a good solution in the searching efficiency. Moreover, the control performance of the controller selected by the PSO-based method is better than that by the GA-based method.

Table 3. Performance Comparison of GA-based method and PSO-based method in the distance control.

Distance control	*Tr*	*IAE*
GA-based method	14	169
PSO-based method	14	153

Table 4. Performance Comparison of GA-based method and PSO-based method in the angle control.

Angle control	*Tr*	*IAE*
GA-based method	4	105
PSO-based method	4	81

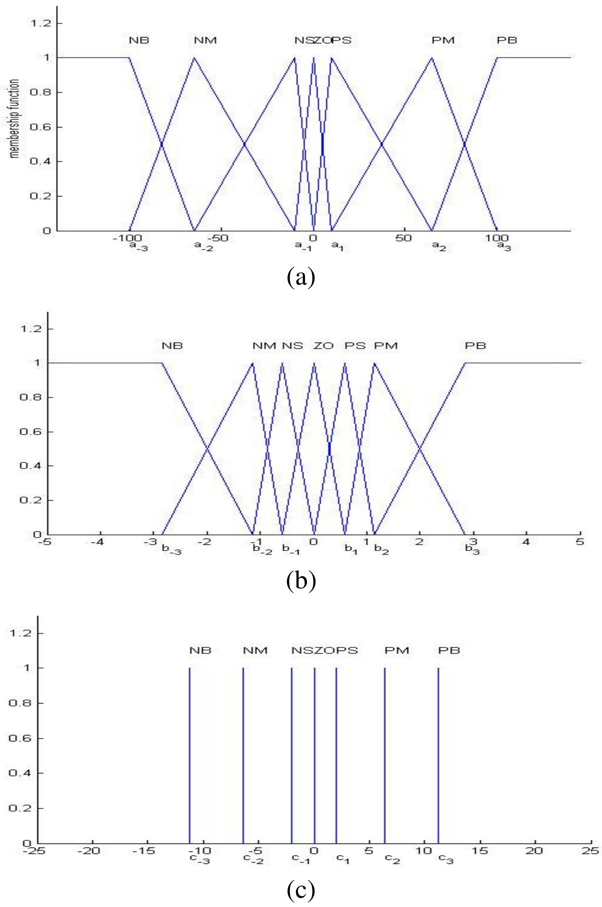


Figure 6. Membership functions of (a) x_1, (b) x_2, and (c) y_1 determined by the proposed PSO-based method.

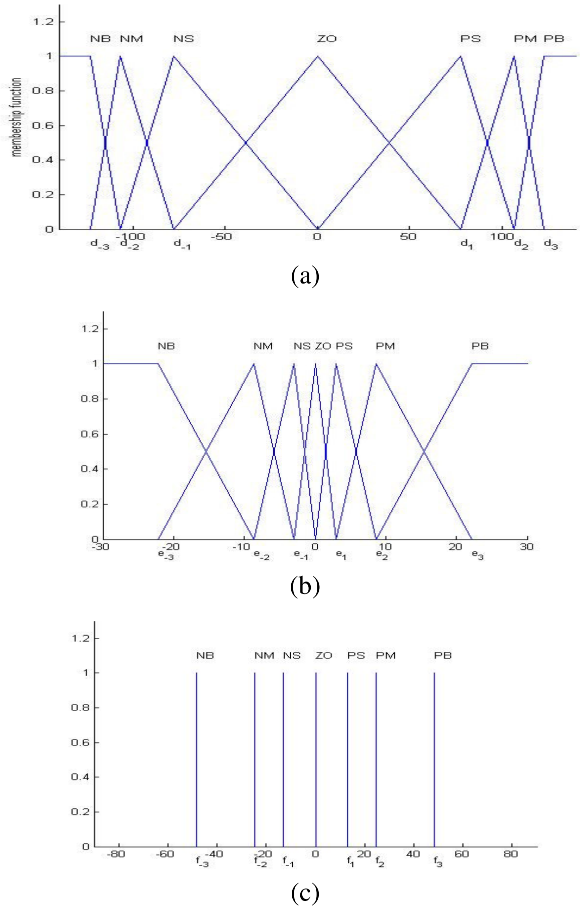


Figure 7. Membership functions of (a) x_3, (b) x_4, and (c) y_2 determined by the proposed PSO-based method.

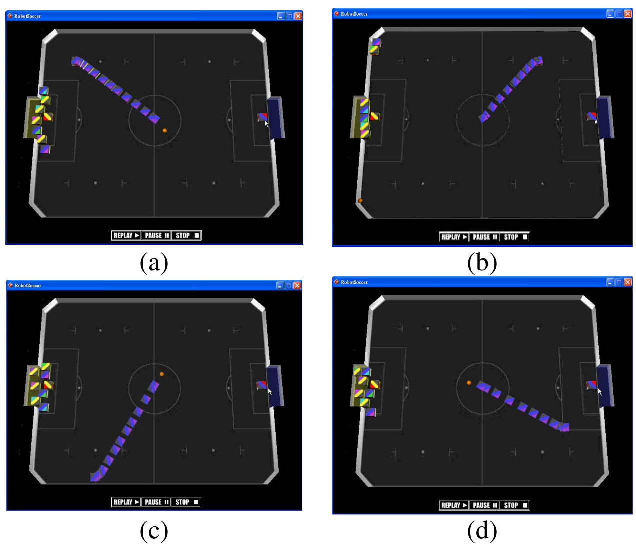
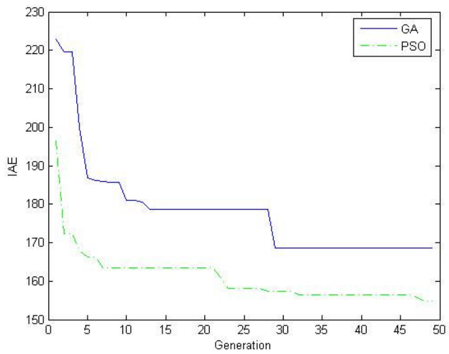
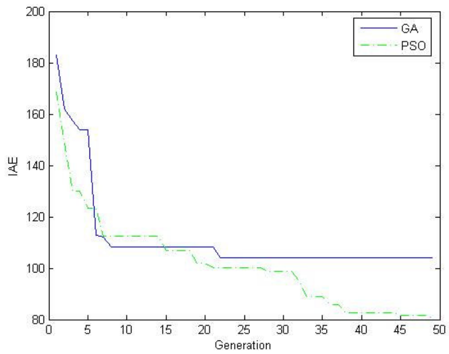


Figure 8. Trajectories of two-wheeled robot controlled by the proposed motion fuzzy controller selected by the PSO-based method and simulated in FIRA 3D robot soccer simulator from four initial postures. (a) (-65, 60, 180), (b) (55, 60, 180), (c) (-55, -90, 180), and (d) (65, -30, 180).

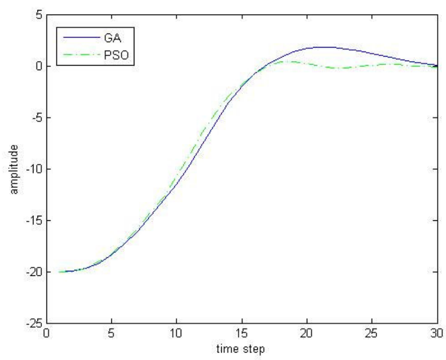


(a)

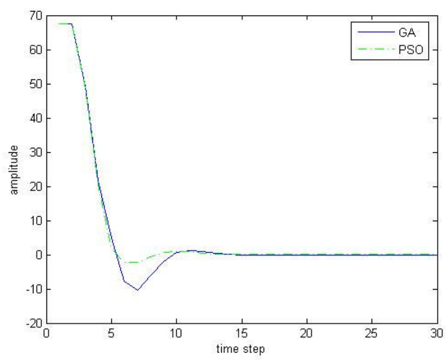


(b)

Figure 9. Searching results of the GA-based method (solid line) and PSO-based method (dash line). (a) Distance control. (b) Angle control.



(a)



(b)

Figure 10. Control results of the controller selected by the GA-based method (solid line) and PSO-based method (dash line). (a) Distance control. (b) Angle control.

5. Conclusions

A PSO-based motion fuzzy controller design method is proposed to determine velocities of the left-wheeled motor and right-wheeled motor of the two-wheeled mobile robot so that the controlled robot can move to any desired position effectively in a two-dimensional space. A PSO-based method is proposed to automatically determine appropriate membership functions of the fuzzy system. Two searching methods are simulated in the 3D robot soccer simulator of FIRA and the control results are analysis by the MATLAB. In the comparison, we can see that both the PSO-based method and the GA-based method have good searching results. But the PSO-based method has a better searching ability than the GA-based method. Moreover, the control performance of the controller selected by the PSO-based method is better than that selected by the GA-based method. The proposed method has been implemented in an object oriented programming method so that the implemented controller has the characteristics of high modularity and portability. In the practical application, the proposed fuzzy controller design method has been successfully applied to control two-wheeled mobile robots for the actual FIRA robot soccer tournament and it also has a good performance.

Acknowledgment

This work was supported in part by the National Science Council of the Republic of China under NSC 96-2218-E-032-001.

References

[1] M. Bowling and M. Veloso, "Motion control in dynamic multi-robot environments," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 168-173, 1999.

[2] W. R. Hwang and W. E. Thompson, "Design of intelligent fuzzy logic controllers using genetic algorithms," *IEEE World Congress on Computational Intelligence*, pp. 2144-1388, 1994.

[3] T. H. Lee, F. H. F. Leung, and P. K. S. Tam, "Position control for wheeled mobile robots using a fuzzy logic controller," *IEEE International Conference on Industrial Electronics Society*, pp. 525-528, 1999.

[4] Y. Lee and S. H. Zak, "Genetic fuzzy tracking controllers for autonomous ground vehicles," *American Control Conference*, pp. 2144-2149, 2002.

[5] C. Lin, A. B. Rad, W. L. Chan, and K. Y. Cai, "A robust fuzzy PD controller for automatic steering control of autonomous vehicles," *IEEE International Conference on Fuzzy Systems*, pp. 549-554,

2003.
[6] F. Solc and B. Honzik, "Modeling and control of a soccer robot," *IEEE International Workshop in Advanced Motion Control*, pp. 506-509, 2002.
[7] C. C. Wong, H. Y. Wang, S. A. Li, and C. T. Cheng "Fuzzy controller designed by GA for two-wheeled mobile robots," *International Journal of Fuzzy Systems,* vol. 9, no. 1, pp. 22-30, 2007
[8] W. L. Xu, S. K. Tso, "Real time self reaction of a mobile robot in unstructured environments using fuzzy reasoning," *Engineering Applications of Artificial Intelligence*, vol. 9, no. 5, pp. 475-485, 1996.
[9] C. C. Chen and C. C. Wong, "Self-generating rule-mapping fuzzy controller design using a genetic algorithm," *IEE Proceedings: Control Theory and Applications,* vol. 149, pp. 143-148, 2002.
[10] C. C. Wong and S. M. Her, "A self-generating method for fuzzy system design," *Fuzzy Sets and Systems*, vol.103, no.1, pp.13-25, 1999.
[11] C. C. Wong and C. C. Chen, "A GA-based method for constructing fuzzy systems directly from numerical data," *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics,* vol. 30, no. 6, pp. 904-911, 2000.
[12] C. C. Wong, B. C. Lin, S. A. Lee, and C. H. Tsai, "GA-based fuzzy system design in FPGA for an omni-directional mobile robot," *Journal of Intelligent & Robotic Systems,* vol.44, no.4, pp.327-347, 2005.
[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," *IEEE International Conference Neural Network*, pp. 1942-1948. 1995
[14] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Sixth International Symposium on Micro Machine and Human Science*, *Nagoya Japan*, pp.39-43,1995.
[15] J. Kennedy, "The particle swarm: Social adaptation of knowledge," *International Conference on Evolutionary Computation*, pp. 303-308, 1997.
[16] Z. L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. on Energy Conversion,* vol. 19, no. 2, pp.384-391, 2004.
[17] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp.58-73, 2002.
[18] F. Ye, C. Y. Chen, and H. M. Feng, "Automatic evolutional clustering-based fuzzy modeling system design," *International Journal of Fuzzy Systems*, vol. 7, no.4, pp. 182-190, 2005.
[19] http://www.fira.net/soccer/simurosot/overview.html

Ching-Chang Wong received a B.S. degree in electronic engineering from Tamkang University, Taipei Hsien, Taiwan, in 1984. He received a M.S. and a Ph.D. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1986 and 1989, respectively. He joined the Electrical Engineering Department of Tamkang University in 1989 and now is a professor. His research interests include fuzzy systems, intelligent control, SOPC design, and robot design.

Hou-Yi Wang received a B.S. and a M.S. degree in electrical engineering from Tamkang University, Taiwan, in 2002 and 2004, respectively. He is currently pursuing a Ph.D. degree in electrical engineering from Tamkang University. His research interests include fuzzy system, strategy design, FPGA design, and evolutional algorithms.

Shih-An Li received a B.S. degree in electrical engineering from National Taipei University of Technology, Taiwan, in 1997. He received a M.S. degree in electrical engineering from Tamkang University, Taiwan, in 2004. He is currently pursuing a Ph.D. degree in electrical engineering from Tamkang University. His research interests include fuzzy system, FPGA design, and genetic algorithms.