

ACA-based Fuzzy Controller Design for Robot Soccer

Juing-Shian Chiou and Kuo-Yang Wang

Abstract

This paper presents the use of an ant colony algorithm combined with a fuzzy logic controller (ACA-FLC) to optimize the movement of a soccer robot. Through ACA-FLC processes, the optimal angle and velocity of a moving robot in the work space were determined. At later stages of the study, more effective movement was obtained from ACA-FLC processes. The experimental scenarios involved a simulated 5-vs-5 soccer game and a MATLAB simulation, in which the proposed system dynamically directed the robot to the target position. The simulation results show that the ACA-FLC was able to choose a better position under any conditions it encountered. The ACA-based method is proposed to determine appropriate membership functions automatically for the fuzzy system so that the controlled robot can effectively move to any desired position in a 2-D space. Following the steps outlined above, the proposed ACA-FLC helped the robot reach the target quickly and effectively.

Keywords: Ant colony algorithm, Fuzzy logic controller, Mobile robot, Robot soccer.

1. Introduction

We utilized the robot soccer system as our test platform since it can be used to fully implement a multi-agent system. The best known associations are the Federation of International Robot Soccer Association (FIRA) and the Robot World Cup Soccer Games and Conferences. Robot soccer requires a high level of understanding of the various engineering domains, such as electronics, computer science, and mechanical engineering [1-6]. There are two kinds of simulation platforms in the FIRA: eleven-versus-eleven and five-versus-five. We chose the five-versus-five simulation platform, which is shown in Figure 1.

To achieve an optimal design for a soccer robot, this study focused on the use of generalized predictive control (GPC) to predict the possible position of the

target at the next sampling time. The optimal speed of the robot was then revised through the application of the combination of a fuzzy controller and the ant colony algorithm. The latter was used to design the obstacle-avoidance routes.



Figure1. Five-versus-five simulation platform.

Generalized predictive control (GPC) has been shown to be particularly effective for the self-tuning control of industrial processes. GPC belongs to the class of model-based predictive control techniques and was first introduced by Clarke and his colleagues in 1987 [7-9]. In this study, we designed a GPC to help the robot quickly predict the position of the target at the next sampling time. The process involves the robot extrapolating the next sampling time target from the time initially required to reach the target in order to decide the route it should follow [10].

On the other hand, the ant colony algorithm is based on the ability of ants to find the shortest route back home. They do not rely on vision, but on a substance called pheromones, which they leave behind them when they pass. Later, when another ant walks by, there is a high possibility that it will choose the path with a high level of pheromones. Therefore, as time goes by, ants will follow the same route back and forth between their nest and the location of food. This theory helps to understand optimization problems a little bit more.

In this study, we first employed the GPC to identify the next possible position of the target, after which we used the fuzzy control machine to determine the speed of both the right and left wheels of the robot. We then introduced the ant colony algorithm to adjust its fuzzy membership functions and thus obtain the optimal result. Finally, we calculated the obstacle-avoidance routes according to the ant colony algorithm to ensure the efficiency and effectiveness of the strategy. Figure 2 shows the flowchart of the system.

Corresponding Author: Juing-Shian Chiou is with the Department of Electrical Engineering Southern Taiwan University, 1 Nan-Tai St, Yung-Kang City, Tainan Hsien, 710, Taiwan, R.O.C.
E-mail: jschiou@mail.stut.edu.tw

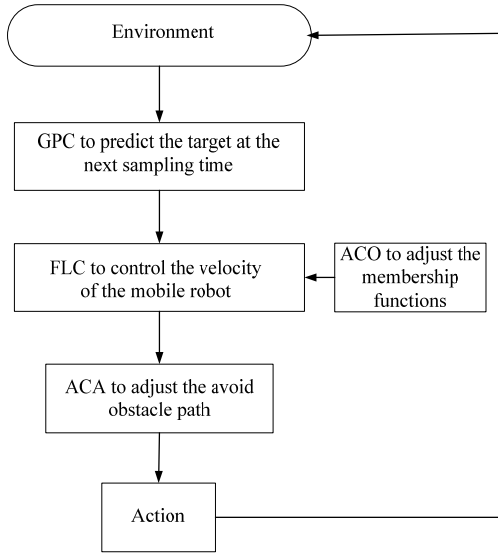


Figure 2. System architecture.

In short, an ACA-based fuzzy controller design is proposed to automatically determine the appropriate membership functions of the fuzzy systems in order to control a soccer robot, so that it moves efficiently in two-dimensional space.

The rest of this paper is organized as follows: In Section 2, we describe the use of the GPC method to identify the target position at the next sample time. In Section 3, the soccer robot is described, and the structure of a fuzzy logic controller of motion is proposed to determine the velocities of its left and right wheels. In the fuzzy logic controller, the inputs of the 2 input-1 output fuzzy system are the distance and the angle of motion between the robot and the target. Section 4 describes the use of the ACA-based method to adjust the fuzzy membership functions. In Section 5, we mainly use the ant colony algorithm to find the obstacle-avoidance route as the robot is moving. In Section 6, some simulations are carried out in the 3D robot soccer simulator of the FIRA and the MATLAB simulator, and the results are presented to illustrate the efficiency of the proposed design. Finally, conclusions are presented in Section 7.

2. Using GPC to predict the target position

Adaptive predictive control machines include a classified on-line structure and control system. The design parameters of the GPC include the autoregressive exogenous classification of on-line model levels and the controlled weighting of the control force. We utilized the current position and sampling time of the target to predict the target position at the next sampling time. We applied the predictive principles of GPC to establish extremely short sampling times, thus making it possible

to predict the next potential position of the target. Finally, the action of the robot was determined. The following steps illustrate the procedure followed.

Step 1: Although this system is nonlinear, we used an extremely short sampling time, and as a result of state transformation during this whole process, we can define this as a linear system.

Step 2: At first, we established the useful conditions of the system: The position of the robot (R_x, R_y) ; the former position of the target (GF_x, GF_y) ; and the current position of the target (G_x, G_y) . The former and current positions of the target were determined on the basis of sampling time T_0 , as shown in Figure 3.

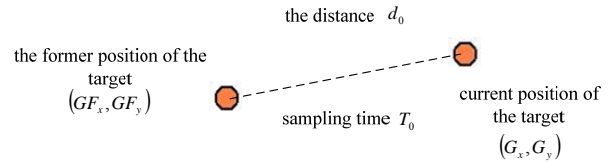


Figure 3. Sampling time

Step 3: We calculated the distance between the former and current positions of the target by means of (1), and we determined the speed of the target V_0 with (2) by using the sampling time T_0 .

$$d_0 = \sqrt{(GF_x - G_x)^2 + (GF_y - G_y)^2} \quad (1)$$

$$V_0 = d_0 / T_0 \quad (2)$$

Step 4: After calculating the velocity of the target V_0 , we designed a GPC by using V_0 , the direction of the target and the sampling time T_0 , which was then used to find the subsequent position of the target (GL_x, GL_y) , as calculated by means of (3) below and illustrated in Figure 4.

$$\begin{cases} d_0 = \sqrt{(G_x - GL_x)^2 + (G_y - GL_y)^2} \\ \frac{GF_y - G_y}{GF_x - G_x} = \frac{G_y - GL_y}{G_x - GL_x} \end{cases} \quad (3)$$

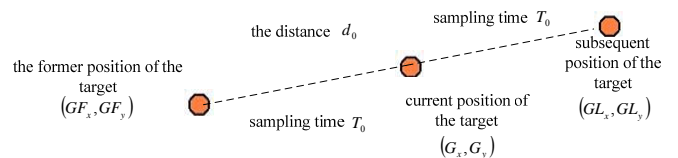


Figure 4. Subsequent position of the target

Step 5: By using (4):

$$T_1 = d / V_c \tag{4}$$

we calculated the time T_1 needed for the robot to reach the target at its central speed V_c , which was calculated with (5) and based on the distance (d) that it had to cover to reach its current position. The equations below show the calculations involved. (V_L is the speed of the left wheel of the robot, and V_R is the speed of the right wheel.)

$$V_c = \frac{|V_L| + |V_R|}{2} \tag{5}$$

Step 6: If T_1 was larger than T_0 , then the robot followed to the next position of the target; if T_1 was smaller than T_0 , the robot proceeded to the current position of the target. By repeating steps 1 to 6, the target could be reached in less time.

3. Motion fuzzy controller structure

In this part, we start with a design for a fuzzy logic controller (FLC) aimed at producing the velocities of the right and left wheels of the robot. Two input parameters of an FLC are distance (d) and angle (ϕ). The former d is the distance between the robot and the goal. The latter ϕ is the orientation of the robot with respect to the straight line path to the goal. Both are shown in Figure 5.

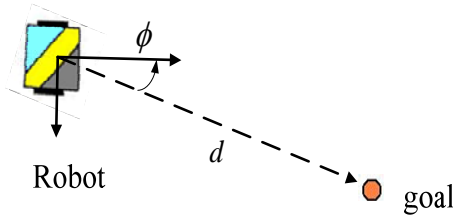


Figure 5. Relationship between d and ϕ .

We represented d , ϕ , V_{LL} and V_{RR} as e_1 , e_2 , y_1 and y_2 , respectively. Afterwards, we set the values of variables e_3 and e_4 ($e_1 = e_3$ and $e_2 = e_4$). We designed two fuzzy controllers to control the velocity of the right and left wheels of the robot. In the first fuzzy controller, e_1 and e_2 are used as the input variables and y_1 as the output variable. In the other fuzzy controller, e_3 and e_4 are used as the input variables and y_2 as the output variable. (In the pairs of equations that follow, the first equation refers to the former, the

second to the latter.)

The fuzzy rules on which were based these fuzzy controllers are described in Table 1 and Table 2, and can be described according to the following equations:

$$R_{y_1}(j_1, j_2): \text{ IF } e_1 \text{ is } A_{(1,j_1)} \text{ AND } e_2 \text{ is } A_{(2,j_2)}, \text{ THEN } y_1 \text{ is } y_{1(j_1,j_2)} \quad j_1, j_2 \in \{-3, -2, -1, 0, 1, 2, 3\} \tag{6}$$

$$R_{y_2}(j_3, j_4): \text{ IF } e_3 \text{ is } A_{(3,j_3)} \text{ AND } e_4 \text{ is } A_{(4,j_4)}, \text{ THEN } y_2 \text{ is } y_{2(j_3,j_4)} \quad j_3, j_4 \in \{-3, -2, -1, 0, 1, 2, 3\} \tag{7}$$

where e_1 , e_2 , e_3 and e_4 are input variables, and y_1 and y_2 are output variables. $A_{(1,j_1)} \in T(e_1)$, $A_{(2,j_2)} \in T(e_2)$, $y_{1(j_1,j_2)} \in T(y_1)$, $A_{(3,j_3)} \in T(e_3)$, $A_{(4,j_4)} \in T(e_4)$ and $y_{2(j_3,j_4)} \in T(y_2)$.

The following term sets were used to describe the fuzzy sets of each input and output fuzzy variables:

$$T(e_i) = \{NB, NM, NS, Z, PS, PM, PB\}, i = 1, 2, 3, 4 \tag{8}$$

$$= \{A_{(i,-3)}, A_{(i,-2)}, A_{(i,-1)}, A_{(i,0)}, A_{(i,1)}, A_{(i,2)}, A_{(i,3)}\}$$

$$T(y_m) = \{NB, NM, NS, Z, PS, PM, PB\}, m = 1, 2 \tag{9}$$

$$= \{y_{(m,-3)}, y_{(m,-2)}, y_{(m,-1)}, y_{(m,0)}, y_{(m,1)}, y_{(m,2)}, y_{(m,3)}\}$$

Each variable has seven linguistic values: Negative Big (NB), Negative Middle (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Middle (PM) and Positive Big (PB). As shown in Figure 6, the triangle membership functions and the singleton membership functions are used to describe the fuzzy sets of input variables and output variables. Based on the weighted average method, the final outputs of these fuzzy controllers can be described by means of (10) and (11).

$$y_1 = \sum_{j_1=-3}^3 \sum_{j_2=-3}^3 w_{(j_1,j_2)} y_{1(j_1,j_2)} \tag{10}$$

$$y_2 = \sum_{j_3=-3}^3 \sum_{j_4=-3}^3 w_{(j_3,j_4)} y_{2(j_3,j_4)} \tag{11}$$

where $w_{(j_1,j_2)}$ and $w_{(j_3,j_4)}$ were determined according to (12) and (13).

$$w_{(j_1,j_2)} = \frac{\min(\mu_{A_{(1,j_1)}}(e_1), \mu_{A_{(2,j_2)}}(e_2))}{\sum_{j_1=-3}^3 \sum_{j_2=-3}^3 \min(\mu_{A_{(1,j_1)}}(e_1), \mu_{A_{(2,j_2)}}(e_2))} \tag{12}$$

$$w_{(j_3,j_4)} = \frac{\min(\mu_{A_{(3,j_3)}}(e_3), \mu_{A_{(4,j_4)}}(e_4))}{\sum_{j_3=-3}^3 \sum_{j_4=-3}^3 \min(\mu_{A_{(3,j_3)}}(e_3), \mu_{A_{(4,j_4)}}(e_4))} \tag{13}$$

When the input data of e_1 , e_2 , e_3 and e_4 are given, y_1 and y_2 can be determined by using (10) and (11). Thus, the left-wheel velocity V_{LL} and the right-wheel velocity V_{RR} can be obtained.

Table 1. Fuzzy rule base of the left-wheel velocity fuzzy controller.

		e_1						
		NB	NM	NS	Z	PS	PM	PB
e_2	NB	MS $Y_{(1,3)}$	NM $Y_{(1,2)}$	NS $Y_{(1,1)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	NM	Z $Y_{(1,3)}$	NM $Y_{(1,2)}$	NS $Y_{(1,1)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	NS	Z $Y_{(1,3)}$	NM $Y_{(1,2)}$	NS $Y_{(1,1)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	Z	PM $Y_{(1,0)}$	Z $Y_{(1,1)}$	NS $Y_{(1,2)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	PS	PS $Y_{(1,-1)}$	Z $Y_{(1,0)}$	NS $Y_{(1,1)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	PM	PM $Y_{(1,-2)}$	Z $Y_{(1,0)}$	NS $Y_{(1,1)}$	Z $Y_{(0,0)}$	PS $Y_{(0,1)}$	PM $Y_{(0,2)}$	PB $Y_{(0,3)}$
	PB	Z $Y_{(1,-3)}$	NM $Y_{(1,-2)}$	NS $Y_{(1,-1)}$	Z $Y_{(0,-1)}$	PS $Y_{(0,-2)}$	PM $Y_{(0,-3)}$	PB $Y_{(0,-4)}$
		Z $Y_{(1,-3)}$	NM $Y_{(1,-2)}$	NS $Y_{(1,-1)}$	Z $Y_{(0,-1)}$	PS $Y_{(0,-2)}$	PM $Y_{(0,-3)}$	PB $Y_{(0,-4)}$

Table 2. Fuzzy rule base of the right-wheel velocity fuzzy controller.

		e_1						
		NB	NM	NS	Z	PS	PM	PB
e_2	NB	Z $Y_{(1,3)}$	NS $Y_{(1,2)}$	NS $Y_{(1,1)}$	NS $Y_{(0,0)}$	NB $Y_{(0,1)}$	NM $Y_{(0,2)}$	NM $Y_{(0,3)}$
	NM	PM $Y_{(1,2)}$	Z $Y_{(1,1)}$	NS $Y_{(1,0)}$	NM $Y_{(0,1)}$	NM $Y_{(0,2)}$	NM $Y_{(0,3)}$	Z $Y_{(0,4)}$
	NS	PM $Y_{(1,1)}$	Z $Y_{(1,0)}$	NS $Y_{(0,1)}$	Z $Y_{(0,0)}$	NM $Y_{(0,1)}$	NM $Y_{(0,2)}$	Z $Y_{(0,3)}$
	Z	PS $Y_{(1,0)}$	PM $Y_{(1,-1)}$	Z $Y_{(1,0)}$	PM $Y_{(0,0)}$	NM $Y_{(0,1)}$	Z $Y_{(0,2)}$	PM $Y_{(0,3)}$
	PS	PB $Y_{(1,-1)}$	PS $Y_{(1,-2)}$	PM $Y_{(1,-1)}$	PS $Y_{(0,0)}$	Z $Y_{(0,1)}$	PM $Y_{(0,2)}$	PS $Y_{(0,3)}$
	PM	PB $Y_{(1,-2)}$	PS $Y_{(1,-3)}$	PM $Y_{(1,-2)}$	PB $Y_{(0,-1)}$	NM $Y_{(0,-2)}$	PS $Y_{(0,-3)}$	PM $Y_{(0,-4)}$
	PB	PB $Y_{(1,-3)}$	PS $Y_{(1,-4)}$	PM $Y_{(1,-3)}$	PB $Y_{(0,-2)}$	NS $Y_{(0,-3)}$	NM $Y_{(0,-4)}$	Z $Y_{(0,-5)}$
		PB $Y_{(1,-3)}$	PS $Y_{(1,-4)}$	PM $Y_{(1,-3)}$	PB $Y_{(0,-2)}$	NS $Y_{(0,-3)}$	NM $Y_{(0,-4)}$	Z $Y_{(0,-5)}$

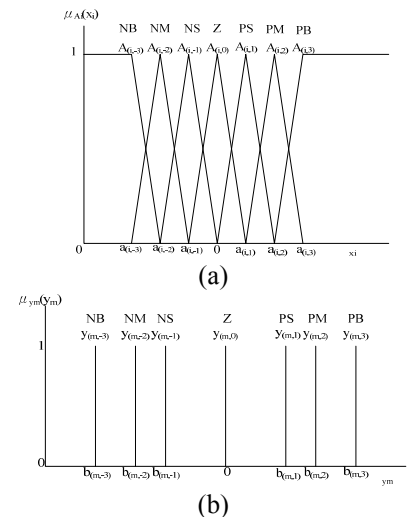


Figure 6. Membership functions: (a) the fuzzy sets for e_1 ; (b) the fuzzy sets for y_m .

4. ACA-based fuzzy controller design method

The main function of the Ant Colony Algorithm is to solve problems in identifying the optimal path to be taken, which is similar to the problem faced by the robot.

As such, when applied to choosing an appropriate path, this algorithm is much more effective than any other and has a shorter merging time. With the velocities generated by the FLC, we determined the maximum velocity of the robot. To this end, we devised the state equation for the robot. The moving velocity of the robot was calculated according to $v = r \cdot \omega$, where v represents velocity. The moving acceleration of the robot was calculated according to $a = r \cdot \dot{\omega}$. In these equations, r is the radius of the robot's wheel (r is 4 cm in this study), and ω is the angular speed of its movement (initial ω_l and $\omega_r = 0$). In what follows, D is the distance travelled by the mobile robot. Accordingly, we were able to calculate the acceleration of the left and right wheels. We defined the mathematical model of the equation of the robot's movements as follows:

$$\left\{ \begin{aligned}
 \dot{\phi} &= \frac{r}{D}(\omega_r - \omega_l) \\
 \dot{x} &= V_{l_x} \\
 \dot{y} &= V_{l_y} \\
 \dot{m} &= V_{r_x} \\
 \dot{n} &= V_{r_y} \\
 \dot{q} &= r \cdot \omega_l \cdot \cos \phi \\
 \dot{w} &= r \cdot \omega_l \cdot \sin \phi \\
 \dot{p} &= r \cdot \omega_r \cdot \cos \phi \\
 \dot{s} &= r \cdot \omega_r \cdot \sin \phi \\
 \dot{\omega}_r &= a_r \\
 \dot{\omega}_l &= a_l
 \end{aligned} \right. \quad (14)$$

Having defined the state variables as follows:

$$x_1 = x, x_2 = y, x_3 = m, x_4 = n, x_5 = V_{l_x}, x_6 = V_{l_y}, x_7 = V_{r_x}, x_8 = V_{r_y}, x_9 = \omega_r, x_{10} = \omega_l$$

we were able to determine the state equation:

$$\begin{aligned}
 & [\dot{x}_1 \ \dot{x}_2 \ \dot{x}_3 \ \dot{x}_4 \ \dot{x}_5 \ \dot{x}_6 \ \dot{x}_7 \ \dot{x}_8 \ \dot{x}_9 \ \dot{x}_{10}]^T \\
 &= [x_5 \ x_6 \ x_7 \ x_8 \ r \cdot a_l \cdot \cos \phi \ r \cdot a_l \cdot \sin \phi \ r \cdot a_r \cdot \cos \phi \ r \cdot a_r \cdot \sin \phi \ a_r \ a_l]^T
 \end{aligned} \quad (15)$$

where

V_{l_x} is a vector of the velocity along the horizontal axis of the robot's left wheel.

V_{l_y} is a vector of the velocity along the vertical axis of the robot's left wheel.

V_{r_x} is a vector of the velocity along the horizontal axis of the robot's right wheel.

V_{r_y} is a vector of the velocity along the vertical axis of the robot's right wheel.

x is a vector of the displacement along the horizontal axis of the robot's left wheel.

y is a vector of the displacement along the vertical axis of the robot's left wheel.

m is a vector of the displacement along the horizontal axis of the robot's right wheel.

n is a vector of the displacement along the vertical axis of the robot's right wheel.

Thus, we identified the optimal vector of the velocity of the left wheel as $\overline{GV}_{LL} = \overline{x_5} + \overline{x_6}$, and that of the right wheel as $\overline{GV}_{RR} = \overline{x_7} + \overline{x_8}$. After that, we changed the vector quantity into a pure quantity and determined the optimal velocity of the left wheel of the mobile robot.

After having determined the velocities based on the fuzzy controller, the seven linguistic items were considered for the input and output variables. This was done to see if these membership functions were symmetrical. 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 were needed to determine all of the membership functions for each variable. Thus, three sets of three center values for each membership function, $\{a_{(1,1)}, a_{(1,2)}, a_{(1,3)}\}$, $\{a_{(2,1)}, a_{(2,2)}, a_{(2,3)}\}$ and $\{b_{(1,1)}, b_{(1,2)}, b_{(1,3)}\}$, were used to determine the input and output membership functions of e_1 , e_2 and y_1 for the fuzzy controller described in (6). Similarly, $\{a_{(3,1)}, a_{(3,2)}, a_{(3,3)}\}$, $\{a_{(4,1)}, a_{(4,2)}, a_{(4,3)}\}$ and $\{b_{(2,1)}, b_{(2,2)}, b_{(2,3)}\}$, were used to determine the input and output membership functions of e_3 , e_4 and y_2 for the other fuzzy controller described in (7).

After we set the fuzzy membership functions, we revised them by using possible questions contained in the ant colony algorithm. First, we set the membership functions $\{a_{(1,1)}, a_{(1,2)}, a_{(1,3)}\}$, $\{a_{(2,1)}, a_{(2,2)}, a_{(2,3)}\}$, $\{a_{(3,1)}, a_{(3,2)}, a_{(3,3)}\}$, $\{a_{(4,1)}, a_{(4,2)}, a_{(4,3)}\}$, $\{b_{(1,1)}, b_{(1,2)}, b_{(1,3)}\}$ and $\{b_{(2,1)}, b_{(2,2)}, b_{(2,3)}\}$, where a and b are the range of the membership function of the fuzzy controller. In this case, a is the input and b is the output. At first, we simply defined the range of the membership function and then calculated the optimal input and output membership function by means of the ACA. Because a domain is defined as the limits of the membership function, we can translate this question as a route on a plain one. On the other hand, if there is a group of ants that want to reach the optimized area, they also have to negotiate the three domains $\{a_{(i,1)}, a_{(i,2)}, a_{(i,3)}\}$. Therefore, the ants have to choose the optimal route among these

three domains. We can explain this by means of Figure 7.

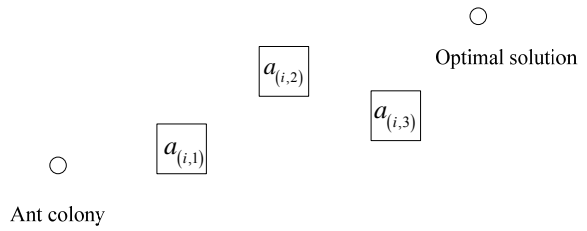


Figure 7. The form of fuzzy rules changed into route one.

After having transformed this question into a route one, we were able to then solve it based on the ant colony algorithm. An ant colony uses many simple agents (ie, ants) to mutually complete its work [11-13]. If we consider $b_i(t)$ as the number of ants at time t in rule i , the ants have to go to four rules, which constitute the optimal solution. As a result, we have $i=1,2,3,4$. However, in terms of the computations involved, i is used to express the domain, $m = \sum_{i=1}^n b_i(t)$ represents the total number of ants, and d_{ij} is the geometric distance from rule i to rule j , which is called the density value of the trace at time $t+1$. Although ants are blind in nature, we give them vision in this system, the so-called visibility, which is calculated according to the multiplicative inverse of the distance, $\eta_{ij} = 1/d_{ij}$. This creates the possibility of choosing among targets, which implies that it is possible for the ant to reach the next rule under the influence of visibility and pheromones. This possibility of choice is expressed by (16). We applied this equation to our system, and the main function of this system is to calculate the probability of a certain route being chosen by the robot.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta}{\sum_{j \in N_i^k} (\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta)} & \text{if } j \in N_i^k \\ 0 & \text{others} \end{cases} \quad (16)$$

Here α and β are the control values, and N_i^k is the domain of the rules unvisited by k ants after passing rule i . The variable α is the factor that triggers pheromones, which shows the relative importance of the route and reflects the accumulation of pheromones during the process of moving. Therefore, if $\alpha = 0$, then the possibility of the ant choosing the next rule depends totally on visibility. The variable β stands for the factor in the expectation equation, which shows the relative importance of visibility. If the value of β is zero, then the decision will be made completely by means of the pheromones. This will lead to a pause in

the solution-finding process, which will finally come up with an optimal route in the same section. The variable d_{ij} shows the distance between two rules. For ant k , the smaller the d_{ij} , the bigger the η_{ij} , for the same $p_{ij}^k(t)$. The variable $\tau_{ij}(t+1)$ stands for the value of pheromones at time $t+1$ along the route from i to j . This is expressed in (17) and (18).

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (17)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (18)$$

Here ρ is the evaporation coefficient, and $\Delta\tau_{ij}^k(t, t+1)$ is the traction density for ant k travelling from rule i to j during time t to $t+1$. In the beginning, τ_0 could be any value. When taking into consideration possibility-choosing questions in the case of an ant wanting to pass three areas in order to reach the optimal target, by applying the method mentioned above, we can come up with a shorter path to the target by comparing all three areas to the optimal target. Thus, it is possible to update the membership functions in the three rules $\{a_{(i,1)}, a_{(i,2)}, a_{(i,3)}\}$, which allows us to achieve optimization.

5. ACA used in obstacle avoidance

In robot soccer games, the choice of obstacle-avoidance path plays an important role. In the hope that our robot would be capable of reaching the target in the shortest possible time, we used the ant colony algorithm (ACA), which is applied to the planning of the obstacle-avoidance path followed by moving objects such as a soccer robot. The ACA uses an adaptive pheromone updating strategy to ensure that the robot reaches the target in the shortest time and follows the best obstacle-avoidance path, as illustrated in Figure 8.

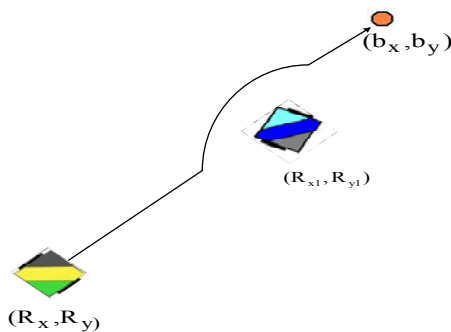


Figure 8. Obstacle-avoidance path.

$$\begin{cases} s(i) = \sqrt{\sum_{l=1}^r \left(\frac{m}{r} - a_l\right)^2} \\ e = \left\lfloor \frac{s(i)}{\max s(i)} (r-1) \right\rfloor + 1 \end{cases} \quad (19)$$

where i and j represent the coordinates of our robot and ball. In this equation, $s(i)$ is the ant colony's concentration of starting points i ; m is the number of ants, and a_l represents the soccer robot following the g paths, which can be expressed as $a_1, a_2, a_3 \dots a_g$.

$\max s(i)$ is the utmost ant colony concentration. In the process, choosing e paths with a high concentration of pheromones causes the ant colony concentration to become higher, which in turn leads to a broader dispersion range at the next step. When $s(i) = \max s(i)$ and $e = r$, the m soccer robot can choose any path, but when the ant colony concentration is at a minimum ($s(i) = 0$ and $e = 1$), the soccer robot will have to choose the best path. This part adopts the objective function to describe the performance of path choice according to (20).

$$\min W = \int_0^L [k\omega_t + (1-k)\omega_f] ds, \quad (0 \leq k \leq 1) \quad (20)$$

where W is the optimized objective function; L is the distance traveled; and ω_t is the time weighted value of the distance to the obstacle. In addition, ω_f is the time weighted value of the distance traveled; k is a coefficient; and ω_f is a function of W . When the objective function has been confirmed, the weighted value of each path that the soccer robot may follow can also be confirmed by means of (21).

$$\omega_g = k\omega_{tg} + (1-k)\omega_{gf} \quad (0 \leq k \leq 1) \quad (21)$$

where w_g is the weighted value of path g ; w_{tg} is the time weighted value of path g ; W_{gf} is the obstacle avoidance weighted value of path g ; and k is a coefficient. If there are n obstacles, $d_{ij}(i, j = 1, 2, \dots, n)$ represents the distance between starting point i and target point j , and $\tau_{ij}(t)$ represents the pheromone concentration between times i and j . We simulated the ant's pheromones by using (22). If there were m ants in all, we used $p_{ij}^k(t)$ to represent the condition according to which the soccer robot transferred the probability from i to j at t .

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{r \in A_k} \tau_{ir}^\alpha(t)\eta_{ir}^\beta(t)} & j \in A_k \\ 0 & \text{others} \end{cases} \quad (22)$$

where A_k represents the obstacle set that the soccer robot may have to avoid in the next step; η_{ij} represents the visibility of the path; and α, β control the relative importance of the relation parameters of $\tau_{ij}(t)$ and η_{ij} . When $\alpha=0$, the soccer robot followed all paths and completed one search. When $\beta=0$, the soccer robot searched about aimlessly. The convergence rate sped up when β was too large. After time N , the soccer robot had found a feasible solution for one planned path, but it may not have been the best solution because the pheromones had changed at this time. Therefore, it was necessary to make an overall change. The amending principle is determined by using (23).

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (23)$$

where $\tau_{ij}(t)$ represents the pheromone concentration of path (i, j) at time t ; ρ represents the pheromone volatility coefficient; $(1-\rho)$ represents the pheromone residual factor; and the value range of ρ is $\rho \in (0, 1)$ in the case of the infinite accumulation of pheromones. In addition, $\Delta\tau_{ij}$ is the variable quantity of pheromones for path (i, j) , as determined by means of (24).

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (24)$$

In this equation, $\Delta\tau_{ij}^k$ represents the pheromone value for soccer robot k . It is just like that of the ant-cycle type, as shown in (25).

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{in this search pass through } i \text{ to } j \\ 0, & \text{others} \end{cases} \quad (25)$$

where Q is a constant, and L_k represents the length of the entire search carried out by robot k .

The following details the procedure for using the ant colony algorithm.

Step 1: Parameter Initialization. At search time $N=0$, set a predetermined search time of NC . Generate m initial solutions at random. Posit that there are s initial solutions following path (i, j) , the total length of which is L^1, L^2, \dots, L^s . Finally, initialize the pheromones of path (i, j) by means of (26), where Q is a constant.

$$\tau_{ij}(0) = \sum_{k=1}^s \frac{Q}{L_k} \quad (26)$$

Step 2: Iterative process. Calculate the distribution range of the ant colony concentration at starting point i according to (27).

$$\begin{cases} e = \left[\frac{s(i)}{\max s(i)} (r-1) \right] + 1 \\ s(i) = \sqrt{\sum_{l=1}^r \left(\frac{m}{r} - a_l \right)^2} \end{cases} \quad (27)$$

Then, calculate the probability of the path choice according to (28).

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta(t)}{\sum_{r \in A_k} \tau_{ir}^\alpha(t) \eta_{ir}^\beta(t)} & j \in A_k \\ 0 & \text{other} \end{cases} \quad (28)$$

Step 3: Update the pheromone concentration for the path according to (29).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (29)$$

Step 4: Repeat Steps 2 and 3 until the ant reaches its target point.

Step 5: Stop the iterative search when one in m ants has already completed its search for the path length and has exceeded the best path length of the previous iteration.

Step 6: Make $N=N+1$, place the ant at the starting point, reset the target point at $N < NC$, and repeat Step 2. Otherwise, output the best path and stop the Algorithm.

6. Simulation results

The PC used to perform the simulations has the following specifications: Pentium IV 3.20G HZ; 1.5GB of RAM; a screen resolution of 800 x 600; Microsoft Windows XP; the ATI X1300-series 3D graphics accelerator with 128MB of RAM; and Direct X 8.0. The parameters in the proposed controller are $\alpha=1$, $\beta=2$, and $\rho=0.6$.

First of all, we compared the differences between SVM-FLC and ACA-FLC (See Figure 9) used to control the speed of the soccer robot. From Figure 9, we can see that the speed of the robot controlled by means of ACA-FLC converged to the optimal speed more quickly than in the case of SVM-FLC.

Figure 10 and Figure 11 demonstrate that the determination of the position at the next sampling time by means of the GPC was more effective and efficient than that done without the estimation provided by the GPC. The touch point in Figure 11 shows that the GPC can truly estimate the next position of the target, and that it allows the robot to get to the target more quickly than the conventional controller (X is the route of the robot; O is the path of the target's movement).

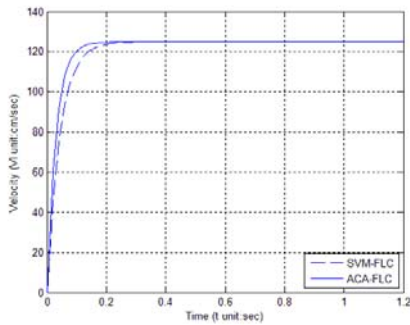


Figure 9. Use of SVM-FLC and ACA-FLC to control the speed of the robot.

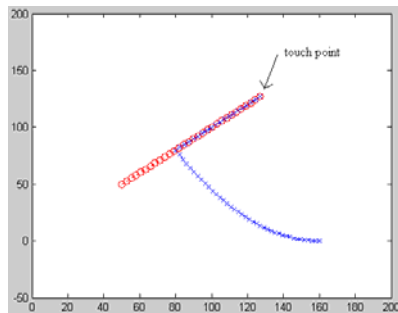


Figure 10. Before using the GPC to predict the next target position (x, y coordinates: inch).

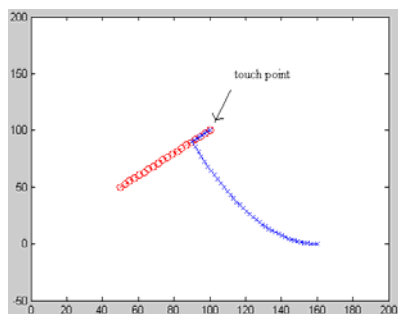


Figure 11. Using the GPC to predict the next target position (x, y coordinates: inch).

In the following part, we simulated the use of the ant colony algorithm to determine the obstacle-avoidance path. By looking at Figure 12 and Figure 13, we can see that when we applied this algorithm to the identification of the obstacle-avoidance path, we obtained a great result.

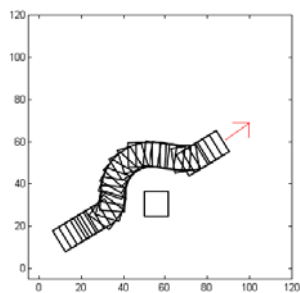


Figure 12. Simulation of obstacle-avoidance path of soccer robot by using MATLAB (x, y coordinates: inch).



Figure 13. Simulation of obstacle-avoidance path of soccer robot by using FIRA simulation.

Next, we mainly imitated the use of ACA-FLC in seeking the path for the controlled robot. In Figure 14 and Figure 15, we can see that after adding ACA-FLC to the design, the path of the robot was the optimal one.

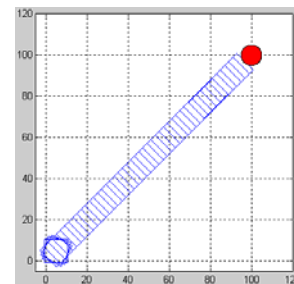


Figure 14. Using ACA-FLC to seek the path of the robot, in the context of a MATLAB simulation (x, y coordinates: inch).



Figure 15. Using ACA-FLC to seek the path of the robot, in the context of a FIRA simulation.

Figure 16 shows the trajectories of the soccer robot when controlled by the proposed ACA-FLC method and simulated with a FIRA 3D simulation.

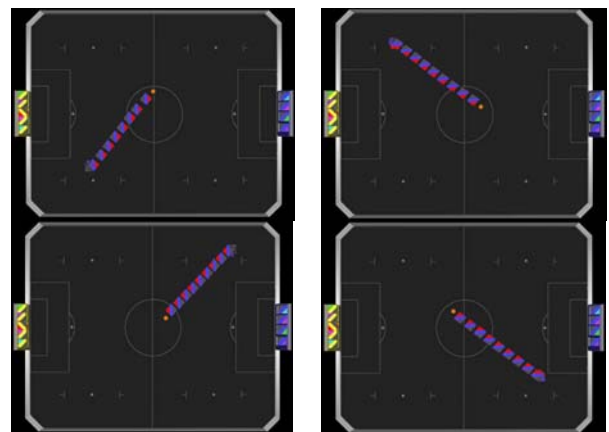


Figure 16. Moving trick.

The membership functions of d , ϕ , y_1 and y_2 , as determined by the proposed ACA-based method, are presented in Figure 17.

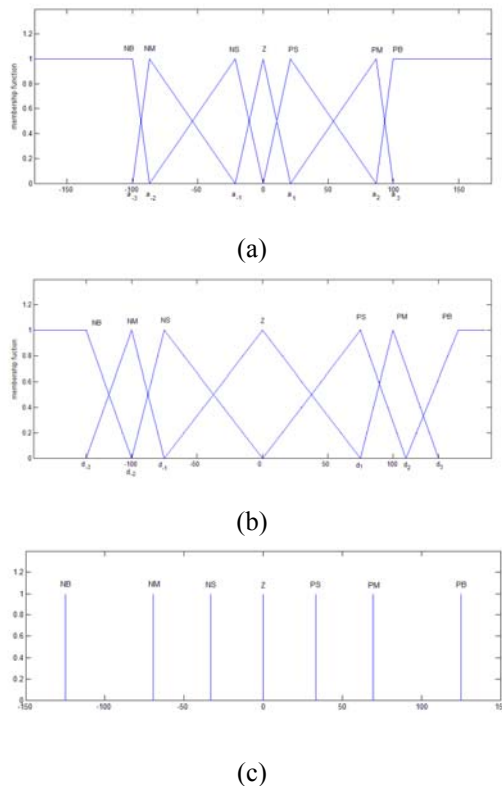


Figure 17. Membership functions of (a) x_1 and x_3 , (b) x_2 and x_4 , and (c) y_1 and y_2 , as determined by the proposed ACA-FLC method.

7. Conclusions

The different kinds of experiments presented in this study confirm the effectiveness of the use of the ACA-FLC with a soccer robot. The advantages and contributions of this study can be listed as follows.

Online learning system: We proposed a method of online learning using the ACA-FLC that performed the immediate adjustment of the velocity and path plan of a soccer robot. In our system, the time required by the ACA to carry out the analysis is about 0.25 seconds.

Dynamic and quick to obtain data on the conditions: In this system, the positions of the mobile robot and target changed quickly. We used the linear programming method to speed up our computations and to identify the target position at the next sampling time.

The results of the experiment presented above show that the method we propose can be effectively applied to a wheeled robot, and the generalized predictive control function we designed can clarify the position of the target at the next sampling time. Also, we used the fuzzy ant colony algorithm to reduce the time required by a robot moving at top velocities to successfully find a path

to its optimal target. At the same time, we used the ant colony algorithm to determine the obstacle-avoidance path.

In the future, we will make further adjustments to the fuzzy ant colony so that it can reach the optimal condition in the shortest possible time. We will add other different algorithms to those mentioned above to an experimental robot to determine their best combination.

Acknowledgment

This work was supported by the National Science Council, Taiwan, Republic of China, under grant number NSC 97-2221-E-218-025.

References

- [1] H. Kitano, M. Asada, I. Noda, and H. Matsubara, "Robocup: robot world cup," *Robotics & Automation Magazine*, vol. 5, no. 3, pp. 30-36, 1998.
- [2] K. S. Hwang, Y. J. Chen, and T. F. Lin, "Q-learning with FCMAC in multi-agent cooperation," *Lecture Notes in Computer Science*, vol. 3971, pp. 599-602, 2006.
- [3] K. S. Hwang, Y. J. Chen, and C. H. Lee, "Reinforcement Learning in Strategy Selection for a Coordinated Multirobot System," *Systems Man and Cybernetics, Part A*, vol. 37, no. 6, pp. 1151-1157, 2007.
- [4] G. Klančar, B. Zupančič, and R. Karba, "Modelling and simulation of a group of mobile robots," *Simulation Modelling Practice and Theory*, vol. 15, pp. 647-658, 2007.
- [5] C. C. Wong, H. Y. Wang, S. A. Li and C. T. Cheng "Fuzzy controller designed by GA for two-wheeled mobile robot," *International Journal of Fuzzy Systems*, vol. 9, no. 1, pp. 22-30, 2007.
- [6] C. C. Wong, H. Y. Wang and S. A. Li "PSO-based Motion Fuzzy Controller Design for Mobile Robots," *International Journal of Fuzzy Systems*, vol. 10, no. 1, pp. 284-291, 2008.
- [7] D. W. Clarke, C. Mohtadi, and P. C. Tuffs, "Generalized Predictive Control-Part 1: The Basic Algorithm," *Automatica*, vol. 23, no. 2, pp. 137-148, 1987.
- [8] D. W. Clarke, C. Mohtadi, and P. C. Tuffs, "Generalized Predictive Control-Part 2: The Basic Algorithm," *Automatica*, vol. 23, no. 2, pp. 149-163, 1987.
- [9] D. W. Clarke, *Advances in Model-Based Predictive Control*, Oxford University Press, 1994.
- [10] J. S. Chiou and K. Y. Wang, "Application of a hybrid controller to a mobile robot," *Simulation*

Modelling Practice and Theory, vol. 16, no. 7, pp. 783-795, 2008.

- [11] M. Dorigo, Gianni Di Caro and Luca M. Gambardella. "Ant Algorithms for Discrete Optimization," *Artificial Life*, vol. 5, no. 2, pp. 137-172, 1999.
- [12] A. Coloni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies," *European conference on artificial life*, pp. 134-142, 1991.
- [13] A. Coloni, M. Dorigo and V. Maniezzo, "An investigation of some properties of an ant algorithm," *the parallel problem solving from nature conference*, pp. 509-520, 1992.



Juing-Shian Chiou received his PhD degree in electrical engineering from National Cheng Kung University, Taiwan, in 2000. Since 1990, he has been with Southern Taiwan University of Technology where he is presently an associate professor in the Department of Electrical Engineering. His research

interests include bilinear systems, singularly perturbed systems, switched linear systems, time-delay systems, fuzzy systems, intelligent control and robot design.



Kuo-Yang Wang received a B.S. and a M.S. degree in electrical engineering from Southern Taiwan University, Taiwan, in 2005 and 2007, respectively. He is currently pursuing a Ph.D. degree in electrical engineering from Southern Taiwan University. His research interests

include fuzzy system, strategy design, ant colony algorithm, support vector machine.