International Journal of Innovative Computing, Information and Control
Volume 9, Number 3, March 2013
ICIC International ©2013 ISSN 1349-4198
pp. 1013–1030

AUTO-GENERATION MODEL/CONTROL FUZZY IMAGE MOBILE ROBOT SYSTEMS DESIGN

Hsuan-Ming Feng[1] and Shiang-Min Jou[2]

[1]Department of Computer Science and Information Engineering
National Quemoy University
No. 1, University Rd., Kin-Ning Village, Kinmen 892, Taiwan
hmfeng@nqu.edu.tw

[2]Department of Electronic Engineering
Xiamen University
No. 422, Siming South Rd., Xiamen 361005, P. R. China
sammy331@pchome.com.tw

Received December 2011; revised April 2012

Abstract. *The evolutional particle swarm optimization (PSO) learning algorithm with self-regulated parameters and an auto-configured fuzzy model machine were applied to efficiently generate the mobile robot control systems. The omnidirectional image sensor was mounted on the mobile robot platform to capture objects surrounding the mobile robot with smart image processing technology to approach the desired traveling path. The generated kinematics mobile robot represents the behavior of the mobile robot in the visual traveling space. The appropriate fuzzy control rules of a mobile robot can be automatically extracted by the direction of the flexibly defined fitness function. The proposed self-learning algorithm can simultaneously avoid obstacles, approach the shortest path, and select the required fuzzy rules numbers. Based on the parameters of the self-generation procedure, the appropriate fuzzy rules were derived to guide the mobile robot toward the desired targets as soon as possible. Six examples of nonlinear mobile robot control problems were applied to demonstrate the adaptability of the self-generated learning algorithm. In the simulated examples, several blocks of various sizes (20, 30, and 40), various locations, and unusual initial and targeted positions were considered to test the adaptation of the learning scheme. Two types of evolutionary PSO learning algorithms were applied to achieve the desired results: one algorithm generates fuzzy rules with an adaptive procedure, and the other algorithm generates fuzzy rules with a random scheme. A comparison of the simulation results of the adaptive PSO (APSO) and random PSO (RPSO) learning algorithms showed that the appropriate mobile robot fuzzy systems were automatically generated by the APSO to form the required fuzzy rules, and detect and escape the obstacles within the desired and shorter traveling path when the initial environments changed.*
Keywords: Fuzzy image systems, Model/Control system, Mobile robots, Particle swarm optimization

1. **Introduction.** Fuzzy rules-based model systems were developed to manage complex, ill-defined, and unexpected nonlinear problems [4,6,17,18]. One of the objectives in fuzzy model generation is to create a fast procedure to accurately describe the behavior of the discussed circumstances. The other considered goal of the fuzzy-based system design is to develop a general and flexible template for solving the wide range of various nonlinear and complex problems. The appropriate generation scheme of the fuzzy model by the training samples is constructed by the following two extracted stages. The first stage is to determine the required number of fuzzy rules which is a crucial procedure in the identification of the fuzzy system structure. The other available stage is called the parameters learning

scheme which determines the fitting values of the parameters that can be adapted for implementation in various mobile robot applications [2,3,7,8,13].

The general production of a mobile robot system consists of sensors, mechanism engine, electric driver, and computer hardware/software integration engineering. Mobile robot systems are widely used in various indoor and outdoor applications [2,7,14]. Because of improvements in modern manufacturing technologies, several smaller and finely embedded components have been rapidly developed over the last few years. Vision-based mobile robots are always integrated with embedded software and hardware devices. Embedded mobile robot systems offer several advantages, such as easy operation, lower power consumption, smaller size, and higher portability, and are widely applied in human life applications. The detection of a particular instance of the particle model in a specific visual image environment without any domain knowledge is a complex problem. Because of the production of precise image devices, several vision-based mobile robot systems were developed to recognize the interested objects in an unknown environment. The challenges in the design of vision-based software are determining the manner in which to acknowledge the anomalous behavior, extract inspected features, analyze the captured image, and select an appropriate control action [2,3,7-9].

In image processing technology, two types of image sensors are used to capture the shape of objects in various environments. The popular omnidirectional image sensors can capture a large scene even if the robot is located in a complex environment. The omnidirectional vision-based sensor can easily detect the spatial and feature dimensions of interest in the surrounding area. The benefit of the omnidirectional image view is that the controller can drive the mobile robots in an arbitrary direction without turning the wheels to track the interested objects and control the motor driver to escape or approach these positions. Another benefit is that the mobile robot can nimbly attain any desired orientation and easily realize its position in the traveling path. Williams II et al. (2002) developed omnidirectional wheels [16] and established dynamic model-based mobile robots with omnidirectional wheels to enable the slipping movement between the driving wheels and the motion surface. Several studies successfully generated vision-based robot systems based on various dynamic training models [2,3,7,8].

Zadeh (1965) was the first to introduce fuzzy logic [19]. Fuzzy systems are known as linguistic rules-based machines, and are widely used to represent human experience. Experts use related domain knowledge to generate autonomous strategies to guide mobile robots to desired targets. Fuzzy inference theory can be successfully used in various applications, especially to solve complex modeling tracking or control problems; however, it is difficult to describe the mathematic model of a robot system. This demonstrates the efficient navigating ability of the robot in approaching the desired targets because of its adaptable self-localization ability. Fuzzy logic is an adaptable theory used to solve the mobile robot model map problem in unknown environments [14,15].

Although successful practical mobile robot applications have been developed using fuzzy logic systems, a number of problems were encountered in approaching the perfect fuzzy rule base to control the mobile robots. One of the main tasks in constructing fuzzy logic systems is to acquire suitable parameters of the fuzzy rules. In the traditional procedure to generate fuzzy rules, the experience of experts and the parameter tuning stage was used by skilled operators in trial-and-error operations. These methods of obtaining the value of parameters are time-consuming. The selected fuzzy rules form the high-dimensional search area, which is a difficult but crucial procedure.

To improve training accuracy, this paper proposes the simple yet efficient evolutionary technique called particle swarm optimization (PSO). The PSO technique was first introduced by Eberhart and Kennedy (1995) [11,12], and was inspired by the social behavior of bird flocking or fish schooling. The concept of the PSO learning stratagem is driven from the scenario of social behavior to solve ill-defined, complex, and global optimization problems. This PSO learning algorithm simulates the behavior of natural creatures as a swarm, and the individual particles are attracted stochastically toward the positions of evaluated optimal performance. In other words, the PSO learning algorithm extracts the required parameter value from the optimal previous experience of the neighbors of the particle from the multidimensional search space. The learning algorithm of PSO, through a metaphor of social interaction, has solved several global optimization mobile robot control problems [2,3,7].

The remainder of this paper is organized as follows. Section 2 introduces the constructed omnidirectional image mobile robot model and fuzzy control structure. Section 3 presents the evolutionary APSO and RPSO learning methods to design the fuzzy model/control system. Section 4 shows three nonlinear mobile robot path tracking problems used to illustrate the effectiveness of the proposed method; and finally, Section 5 offers a conclusion.

2. Omnidirectional Image Mobile Robot Model and Fuzzy Control Structure Design.

The design model of the fuzzy control system is proposed using the omnidirectional image mobile robot. Its relational flowchart is shown in Figure 1. The robot platform contains the main motor driver and robot machine. The main motor driver is a control engine that directly regulates the moving speed and rotating angle of the robot machine when it receives the available fuzzy control command. The omnidirectional image sensor captures the 360° view of the targets and obstacles from the surrounding environment of the mobile robot system. Based on the proposed image processing, the identified target and obstacle images are mapped into the required information of the kinematics model to describe all of the behaviors of the mobile robot. The evolutional PSO

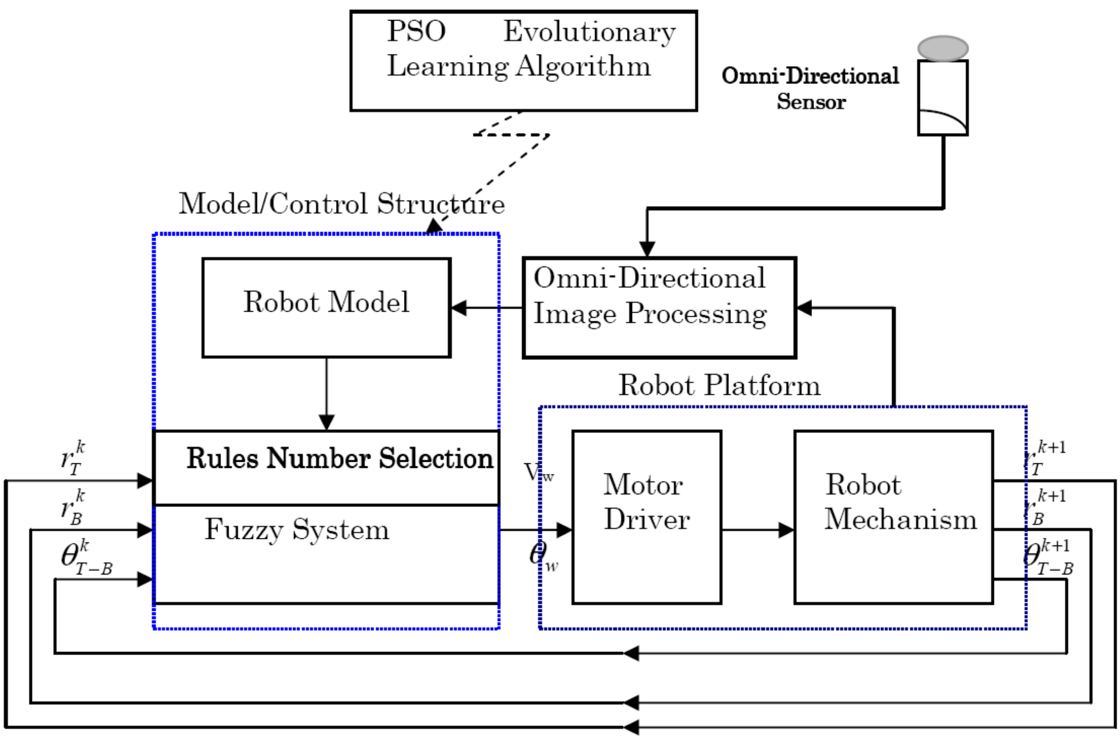

Figure 1. Auto-generation model/control fuzzy image mobile robot systems

algorithm, with the specific purpose of fitness function, was adapted to achieve the appropriate parameter set of the fuzzy system to control the movement of the mobile robot. The objective of the PSO learning algorithm is to minimize the traveling path between the initial robot and the target positions and avoid collision of the defined blocks. The appropriate fuzzy control rules were self-generated by the evolutionary PSO-based learning schemes, which acted as the prototype of the mobile robot in the discussed environments.

In the image processing stage, the simple yet efficient "direct transforming concept" was used to convert the 3D circled-type image into a 2D panoramic view [7]. The omnidirectional caption was performed on the vision-based circled distribution objects to form the environment of the mobile robot, and the view of all objects was subsequently transferred into a panoramic space. The processed panoramic images demonstrated the placement from the scene of the objects. Therefore, this image information is used to detect the interested article using the proposed geometry pattern reorganization method. In the color image processing stage, the 2D color image by the direct transforming method consisted of three channels, that is, red, green, and blue (RGB). The R, G, and B channel-values of the original image are the main color components to represent the true image color in the real world. The HSV space, denoted as the hue, saturation, and bright values, is another popular method to recognize the image pattern. In the HSV space, the effect of the bright value (V) is ignored to distinguish the color of the interested object. It was more efficient than the RGB area to recognize the objects in the other experiment results [10]. The color space transformation formulae from the RGB to HSV domain were obtained from the previous study [7]. After HSV transformations are completed in the captured environment images, the presented colors for the block and targets objects are identified by the (H, S) locations value in the S-H plane. In the HSV color space, the H and S element values are determined to calculate the related angle and the distance between the interested points and marks. The transformed HSV values are used to determine the discussed image objects. Their own colors for the destination and block tracking objects are identified by their (H, S) located positions. In the image morphology stage, the image processing concept was applied to improve the image quality and rebuild the contour of the object accurately. The erosion and dilation technologies were applied to reduce the effect of image flat zones. Two fundamental operations, openings and closings, were used to eliminate the small pellet and reconstruct the completed region to identify the target and obstacle objects in the image training patterns. The openings operation breaks the narrow connection and cleans the small outlier of the objects. The closings operation joins the narrow broken parts and collects a thin gap area to mend the small holes in the inferior image zone. In addition, the small gap of the shape is refilled in this fixed image procedure. The destination and block size determine when the required one-opening-one-closing image procedure is complete.

Based on the developed omnidirectional image processing stage, the extracted objects of a mobile robot, that is, the destination and block in its workspace, can be used to build the visual maps of the mobile robot. The proposed placement plot of the mobile robot system is shown in Figure 2. The locations (x_d, y_d), (x_b, y_b), and (x_r, y_r) are coordinates for the destination, block, and robot, respectively. The kinematics of the mobile robot model was defined by the following formulae:

$$r_d = \sqrt{(x_d - x_r)^2 + (y_d - y_r)^2} \tag{1}$$

$$r_b = \sqrt{(x_b - x_r)^2 + (y_b - y_r)^2} \tag{2}$$

$$\theta_d = \cos\left(\frac{\vec{X}_{dr} \cdot \vec{Y}_{dr}}{\left|\vec{X}_{dr}\right| \cdot \left|\vec{Y}_{dr}\right|}\right)^{-1} \tag{3}$$

$$\theta_b = \cos\left(\frac{\vec{X}_{br} \cdot \vec{Y}_{br}}{\left|\vec{X}_{br}\right| \cdot \left|\vec{Y}_{br}\right|}\right)^{-1} \tag{4}$$

$$\theta_{d-b} = \theta_d - \theta_b \tag{5}$$

where r_d and r_b are the distances between the robot and destination and the robot and block, respectively. θ_d and θ_b are the corresponding angles of the mobile robot for the destination and block in the y-axis, respectively. $\vec{X}_{dr}$ and $\vec{Y}_{dr}$ are vectors, which are calculated from the robot to destinations with respect to the x-axis and y-axis, respectively. $\vec{X}_{br}$ and $\vec{Y}_{br}$ are the distance vectors between the robot to blocks for the x-axis and y-axis, respectively. The $|\,|$ symbol represents the vector length. The proposed robot model by previous mathematical formulae is shown as a visual image map in Figure 2.

In the designed fuzzy image mobile robot system, three input variables (r_d, r_b, θ_{d-b}) and two output variables (θ_w, v_w) were applied to construct the fuzzy inference structure. θ_w and v_w are the turn rates in the angle to the driver of the motor wheels and the traveling line-speed of the robot, respectively.

The mobile robot speed in the current time step k is denoted as v_w^t, and its turn-rate angle is set to θ_w^t. The moving state of the mobile robot with the smaller increasing time

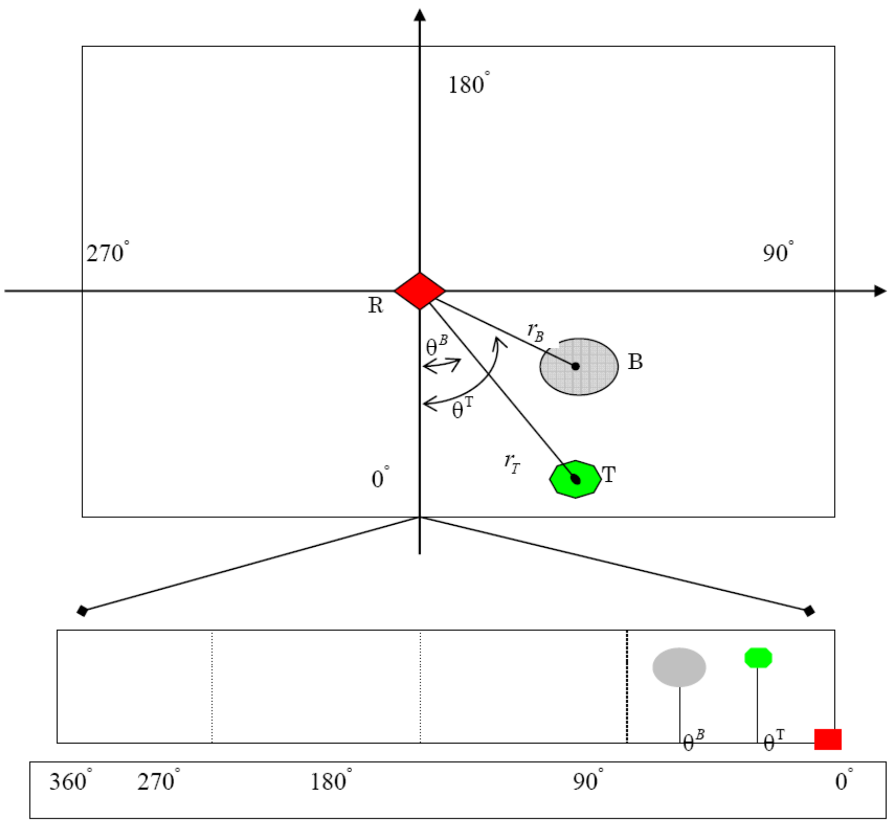

Figure 2. The omnidirectional and Cartesian coordinate mapping plane system

interval (Δk) at the next time step ($k+1$) is described by

$$x_r^{k+1} = x_r^k + v_w^k \bullet \Delta k \bullet \cos(\theta_w^k) \tag{6}$$

$$y_r^{k+1} = y_r^k + v_w^k \bullet \Delta k \bullet \sin(\theta_w^k) \tag{7}$$

Based on (1)-(7), the entire kinematics in the coordinate x-y space can represent the motions of the mobile robot. The fuzzy image model/control system was self-generated by the APSO and RPSO learning algorithm. The details are presented in the following section.

3. **Evolutionary APSO and RPSO Learning Fuzzy Model/Control System Design.** Three variables (r_T, r_B, θ_{T-B}) were applied as a whole N-dimensional vector to form the inputs of the fuzzy system. If $\boldsymbol{X} = (x_1, x_2, \ldots, x_N)$ is denoted as the N-dimensional vector, which is regarded as the premise part of the constructed fuzzy system, the developed fuzzy control rules can be illustrated as follows:

$$\mathrm{R}^{(i)}\text{: IF } \boldsymbol{X} \text{ is } \mathrm{MB}_i \text{ THEN } \boldsymbol{Y} \text{ is } y_i,\ i = 1, 2, \ldots, \mathrm{MUM}, \tag{8}$$

where MUM is the fuzzy rules number and MB_i is the ith fuzzy membership function with respect to the input vector ($\boldsymbol{X}$). The real value y_i is considered the consequent part of the fuzzy system. In this study, the mathematical function of the fuzzy membership function was developed in the following formula:

$$\mathrm{MB}_i(\boldsymbol{X}) = \exp\left(-\left(\frac{(x_1 - c_{i1})^2}{\delta_{i1}^2} + \cdots + \frac{(x_n - c_{iN})^2}{\delta_{iN}^2}\right)\right). \tag{9}$$

A hyper-ellipsoid type function occurred in the N-dimensional search space. The parameters set $(c_{i1}, c_{i2}, \ldots, c_{iN})$ is the central position of the membership functions, and (δ_{ij}) is the actual length of the jth principal axis in the hyper-ellipsoid space. In the designed fuzzy system, the consequent part (y_i) was denoted as a single real value. An appropriate combination of a parameters set (c_{ij}, δ_{ij}, and y_i) was efficiently regulated by the proposed APSO and RPSO algorithms.

In this study, the simple defuzzifier with the weighted average operation was used to convert the fuzzy domain into the real value. Although the firing process for the premise part of the ith rule is justified, the output value of the fuzzy system ($\overline{Y}$) can be calculated using (10).

$$\overline{Y} = \frac{\sum_{i=1}^{m} \mathrm{MB}_i(\boldsymbol{X}) \bullet y_i}{\sum_{i=1}^{m} \mathrm{MB}_i(\boldsymbol{X})}. \tag{10}$$

The contour of the membership function $\mathrm{MB}_i(\boldsymbol{X})$ was approached to present the superior behavior of mobile robots. To provide optimal estimates of the parameters set in generating the kinematics of the mobile robot, the parameters selection problems can be solved using the evolutional APSO and RPSO learning algorithms to derive the appropriate fuzzy model/control mobile robot system.

The behavior of natural creatures is similar to swarm intelligence. The main objective of a swarm artificial system is to observe the manner in which natural creatures act as a swarm. The PSO simulates the swarm models inside a computer computation to yield the optimal characters among the old population. At every learning cycle, the position and velocity of each particle are dynamically regulated by the attraction and direction of the current optimal swarm intelligence values. The personal best, called Pbest, is the optimal solution of every particle, which is in the current state. The other best is called Gbest, which is obtained by the evaluated overall optimal value from all particles in the populations. At the iteration learning step, the velocity of the particle is regulated by the relative optimal values of Pbest and Gbest. The new velocity of each particle is updated using the following formula:

$$\begin{aligned} v_{Np}(k+1) = {} & \tau \cdot v_{Np}(k) + \beta_1 * \mathrm{rand}() * \left(p_{best_{Np}}(k) - Y_{Ni}(k)\right) \\ & + \beta_2 * \mathrm{rand}() * \left(G_{best_{Np}}(k) - Y_{Np}(k)\right) \end{aligned} \tag{11}$$

Therefore, the new particle position is regulated by

$$Y_{Ni}(k+1) = Y_{Ni}(k) + v_{Ni}(k+1) \tag{12}$$

where N and p are the number of dimensions and particles, respectively. k uses the current state, $k+1$ represents the next time step, and β_1 and β_2 are the constant learning rates, which are selected by the designer.

The disadvantage of the traditional fuzzy system is that a membership function number must be defined in advance. In this study, a perfect fuzzy rules number tuning machine was achieved based on the special fitness function design. The fuzzy rules numbers regulation machine allows the PSO learning algorithm to simultaneously determine the proper fuzzy rules number and the optimal parameters for approaching the appropriate fuzzy image mobile robot system. The tuning scheme for approaching the perfect fuzzy rules number using two selection procedures is described as follows.

Let L_j and U_j be the lower and upper bounds of fuzzy rules numbers size, respectively, with regard to the jth particle. The range of L_j and U_j were designed based on the complexity of the optimal search problem. The number list γ_j, where $j = 1, 2, \ldots, p$, was generated by the random procedure, which was determined between the interval $[L_j, U_j]$. The corresponding values γ_j for the fuzzy rules of the jth particle were determined using this procedure. The previous extraction of the active fuzzy rules number was based on the random procedure. $\gamma_j = [5, 6, 2]$ indicated that the remaining active fuzzy rules for Particle 1, Particle 2, and Particle 3 were 5, 6, and 2, respectively. The number of working membership functions for every particle can be determined through this simple and efficient random search scheme. To avoid the problem of the varying lengths of each particle, the random selection operation ensures the same vector length [5]. In the random selection procedure, the largest value from the irregular number list, γ_j (which is 6 in this example), is first selected. Subsequently, the selection value is regarded as the active number of fuzzy rules for all particles. The RPSO learning algorithm with the random selection machine was used to select the fuzzy number γ_j.

$$\gamma_j = \mathrm{RND}\{L_j + (U_j - L_j) * \mathrm{rand}()\}, \tag{13}$$

where RND is the round operator, which generates $\gamma_j \in \{L_j, L_j+1, \ldots, U_j\}$, and rand() is the uniformly distributed random number in the interval $[0, 1]$. The other APSO learning algorithm with the adaptive selection scheme in the fuzzy rules number (NUM) decision was formulated by

$$NUM = \sqrt{\frac{\sum\limits_{j=1}^{sw} (\gamma_j - \overline{\gamma_j})^2}{N-1}} * \mathrm{rand}(), \tag{14}$$

where $\overline{\gamma_j} = \frac{1}{N} \sum\limits_{j=1}^{N} e^{\frac{-\gamma_j}{2\sigma^2}}$. The adaptive selection method can dynamically and smoothly fit the sudden change of the fuzzy rules. The PSO leaning formulae in (11) and (12) can be successfully executed because of the available calculation for the same number of vectors of the membership functions in each particle. The objective of this PSO leaning algorithm is to achieve the appropriate fuzzy model/control systems and select the proper parameters of the fuzzy rules. Let the parameters set $R_j^* = \{\gamma_j, c_{ij1}, c_{ij2}, \ldots, c_{ijn}, \delta_{ij1}, \delta_{ij2}, \ldots, \delta_{ijn}, w_{ij}, 1 \le i \le m\}$ be used to construct the fuzzy model/control system. The developed fitness function FIT(.) with (15) was defined by the PSO learning algorithm to find near optimal solutions, which was performed by achieving the highest fitness value. Let R_i represent the parameters set of the fuzzy mobile robot system in the search space P. The selected FIT (R_i) was denoted as the evaluated fitness value of the input parameter R_i

$$FIT(R_i) = \frac{25}{RMSE} * OB * \frac{7}{NUM}. \tag{15}$$

where the RMSE is considered the root mean square errors of the length between the desired target and initial robot position. OB represents the robot tracking state, which is selected as 1 when the moving trace of the mobile robot does not collide with the block. Otherwise, the other state of OB changes to zero when the mobile robot cannot successfully move through the obstacle in the traveling path. The positive NUM value is denoted as the actual fuzzy rules number. From the discussed elements of the fitness function, the goal of PSO is to maximize the fitness function value, that is, to minimize the RMSE to successfully move through the obstacle and select the required lower fuzzy rules number. The evolutionary PSO-based learning algorithm with the two fuzzy rules number selection machine was used to generate the desired fuzzy model/control system, according to the following steps.

Step 1: Let the maximal iteration number be G and initialize $g = 0$. Select the PSO learning rate (β_1, β_2). Randomly construct the initial populations, which contain the initial particles R_j^* and the fuzzy rules number list, γ_j, at $g = 0$.

Step 2: Select the appropriate fuzzy rules number to form the initial structure of the fuzzy system. Each particle swarm with respect to the jth γ_j is described in the following form:

$$R_j^* = [\gamma_j \quad R_j],$$

where γ_j is the current active fuzzy rules number using both selection methods.

Step 3: Construct the active fuzzy model/control robot system with the jth individual R_j^* using the respective γ_j list.

Step 4: Regulate the parameters using the proposed PSO learning Formulae (11) and (12) to derive the appropriate fuzzy model/control system.

Step 5: Calculate the fitness value for each particle according to the defined fitness function Formula (15).

Step 6: Compare each fitness value with the Pbest to select the new Pbest. Renew the Gbest, which contains the global best fitness value from the Pbest.

Step 7: $g = g + 1$.

Step 8: If $g = G$, exit; otherwise, return to Step 3.

Step 9: The value of the optimal particle, $R_j^* = [\gamma_j \quad R_j]$, is selected as the proper parameter values to develop the desired fuzzy model/control system to obtain the desired robot traveling path.

4. **Case Studies and Illustrated Examples.** To demonstrate the efficiency of the proposed auto-generation PSO learning method, three types of non-linear mobile robot control problems are presented in this section.

Example 4.1. ***Same initial, block, and target positions for different block size cases:*** *In this case study, we used the fuzzy image model system as an identifier to represent the behavior of the mobile robot model. The near-optimal parameters of the fuzzy model/control system were automatically developed by the evolutional PSO learning algorithm. Let PSO swarm sizes = 30; generation = 50, and* $\beta_1 = \beta_2 = 1.25$. *If the adopted* *fuzzy system controller is provided with 10 maximum rules at the start, the two-input and three-output variables of the fuzzy model/control system would contain 90 parameters in every particle. The parameters set* $R_j^* = \{\gamma_j, c1_{ij1}, c2_{ij2}, c3_{ij}, \delta1_{ij}, \delta2_{ij}, \delta3_{ij}, y1_{ij}, y2_{ij}; 1 \leq i \leq 10; 1 \leq j \leq 30\}$ *must be efficiently selected from the search space. In simulated cases, the robot moved from the original* $(20, 20)$ *position to the target location* $(80, -60)$*. The block position was located at* $(40, -20)$ *for the x-axis and y-axis, respectively. The diameters of this block size were set to 20, 30, and 40 values for the adapted experiments. Based on the objective of the fitness function, only three selected fuzzy rules were required for the fuzzy model/control system. It can drive the mobile robot from the initial position to the target point without touching the arranged obstacle in the traveling path. Selection of the fuzzy rules number was also performed in this training cycle. The simulation results for the traveling example with block size = 20 is shown in Figure 3. The simulation results for the adaptive and standard fuzzy rules number selection machine are represented by the magenta dashed line and blue solid line, respectively. Figures 3(a) and 3(b) show the time response of the line speed and turning angle for the controlled mobile robot platform in the testing phase, respectively. Figure 3(c) shows the moving paths of the mobile robot in the nonlinear traveling environment to dynamically overcome the assigned obstructer. These simulations showed that the mobile robot can be driven by the fuzzy model/control system to gradually and smoothly move into the desired target. The fuzzy rules numbers were automatically selected by the developed adaptive and random search machine. The fuzzy rules of the mobile robot were developed using the specific fitness function, which caused the robot platform to approach the desired target within the smaller RMSE value and safely pass through the assigned obstacle. The optimal fitness value with respect to the iteration number is shown in Figure 3(d). This trace demonstrates that the selected parameters of the fuzzy model/control system by the proposed adaptive selection machine achieves superior solutions with the comparison of two fitness values. Figures 3(e) and 3(f) show the dynamic response for the related selected fuzzy rule number and the displayed RMSE value, respectively. The simulation results demonstrate that the specific fitness function achieved the multiple purposes of approaching the minimal fuzzy rules number and smaller RMSE without touching the defined blocks. The robot was set in the same initial and target positions in the other two examples. We assigned the block at the same* $(40, -20)$ *positions, but changed the block sizes to 30 and 40 for the next two simulations, as shown in Figures 4 and 5. The self-constructed fuzzy mobile robot model/control system can realize the captured omnidirectional image patterns to detect the position of blocks of different size and safely move through these obstacles. The required fuzzy rules number was only 3 in these simulated cases. Based on the simulated comparisons of Figures 3-5, the adaptive learning-based rules selection machine quickly tracked the desired path and achieved smaller RMSE results than the random selection procedure for three block size cases (20, 30, and 40). The performance comparisons for these three cases are shown in Table 1. In the simulated result of Case 1, the optimal selected fuzzy rules number was 3, and the RMSE for APSO and RPSO were 2.0187 and 5.9234, respectively. In Case 2, the appropriate fuzzy rules number was 3, and the RMSE for the APSO and RPSO was 2.5393 and 5.3299, respectively. Case 3 demonstrated that the APSO can quickly arrive at the target.*

Example 4.2. ***Different initial positions and block size cases:*** *In the second example, two different initial positions and block size cases were used to demonstrate the robust capability of the evolutionary PSO learning algorithm. In Case 4, the original conditions of the testing environment for the starting position of the mobile robot were changed to* $(-80, 40)$*, the target position was reallocated at* $(-20, -50)$*, and the block position was assigned to* $(-60, -10)$*. Two time responses of the line speed and the rotating*

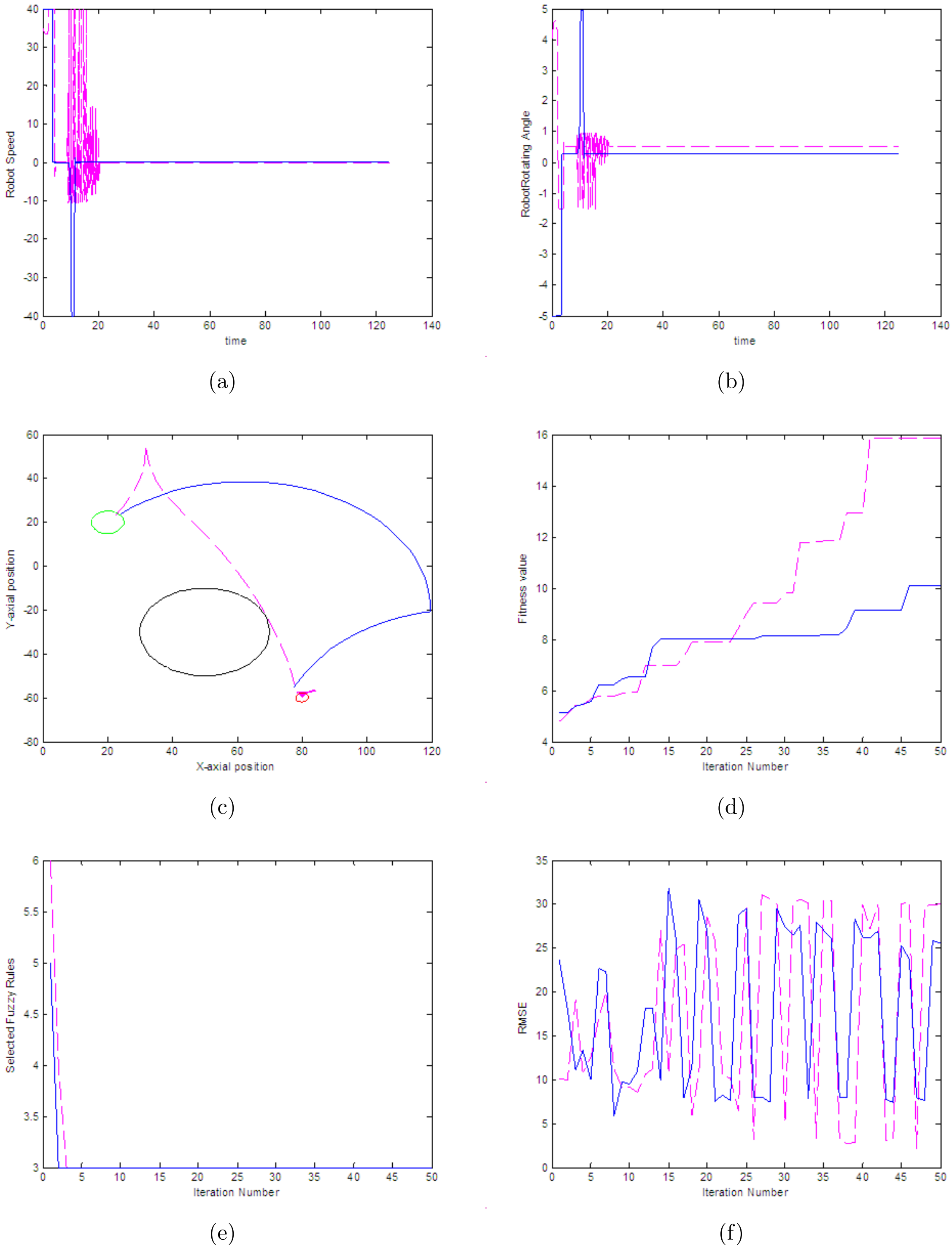

FIGURE 3. Simulation of the first illustrated example. Robot initial position (20, 20), target position (80, −60), block position (50, −30) and block size = 20. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) selected rules number in the respective iteration number and (f) RNSE response for the related iteration number.

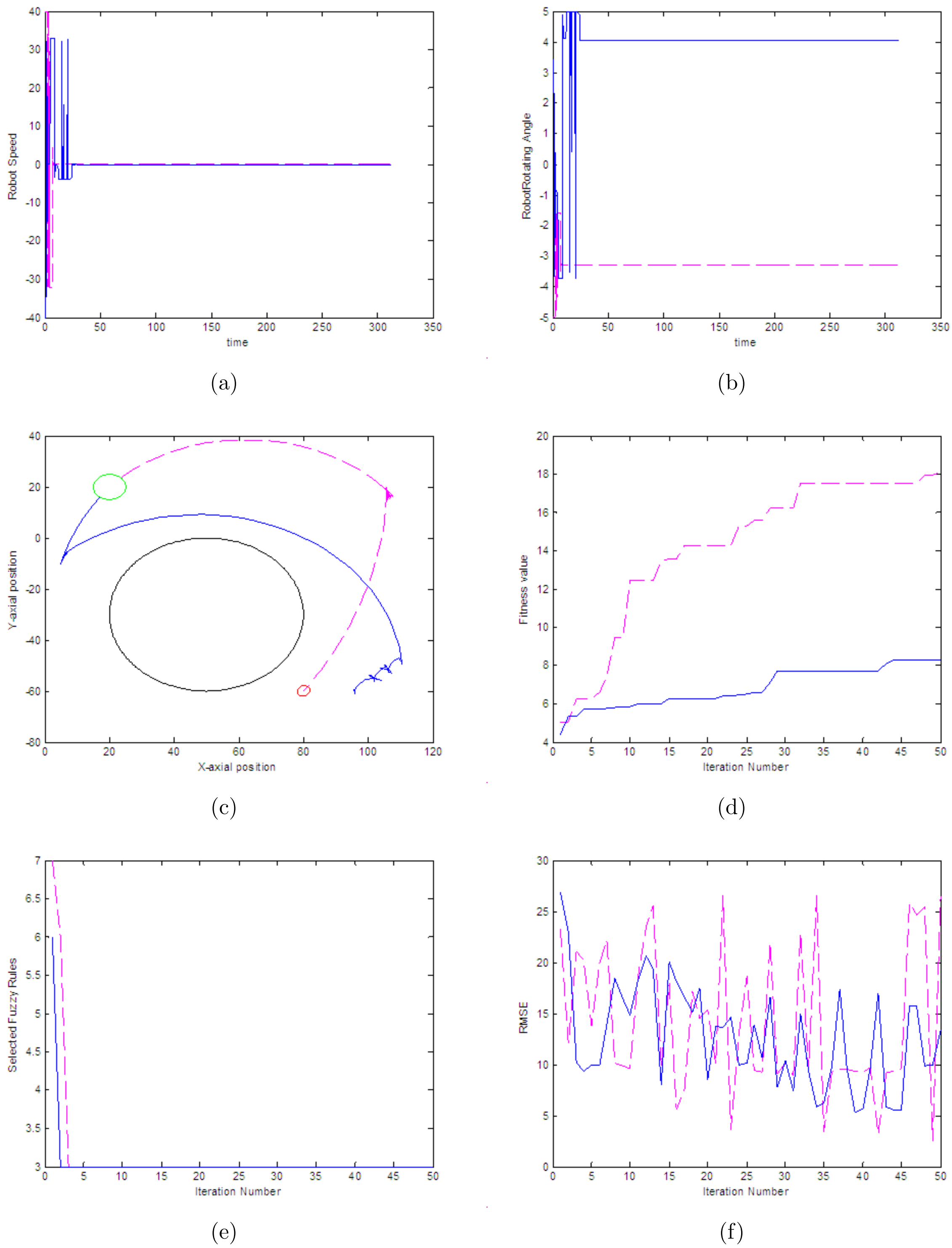

FIGURE 4. Simulation of the first illustrated example. Robot initial position $(20, 20)$, target position $(80, -60)$, block position $(50, -30)$ and block size $= 30$. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) selected rules number in the respective iteration number and (f) RNSE response for the related iteration number.

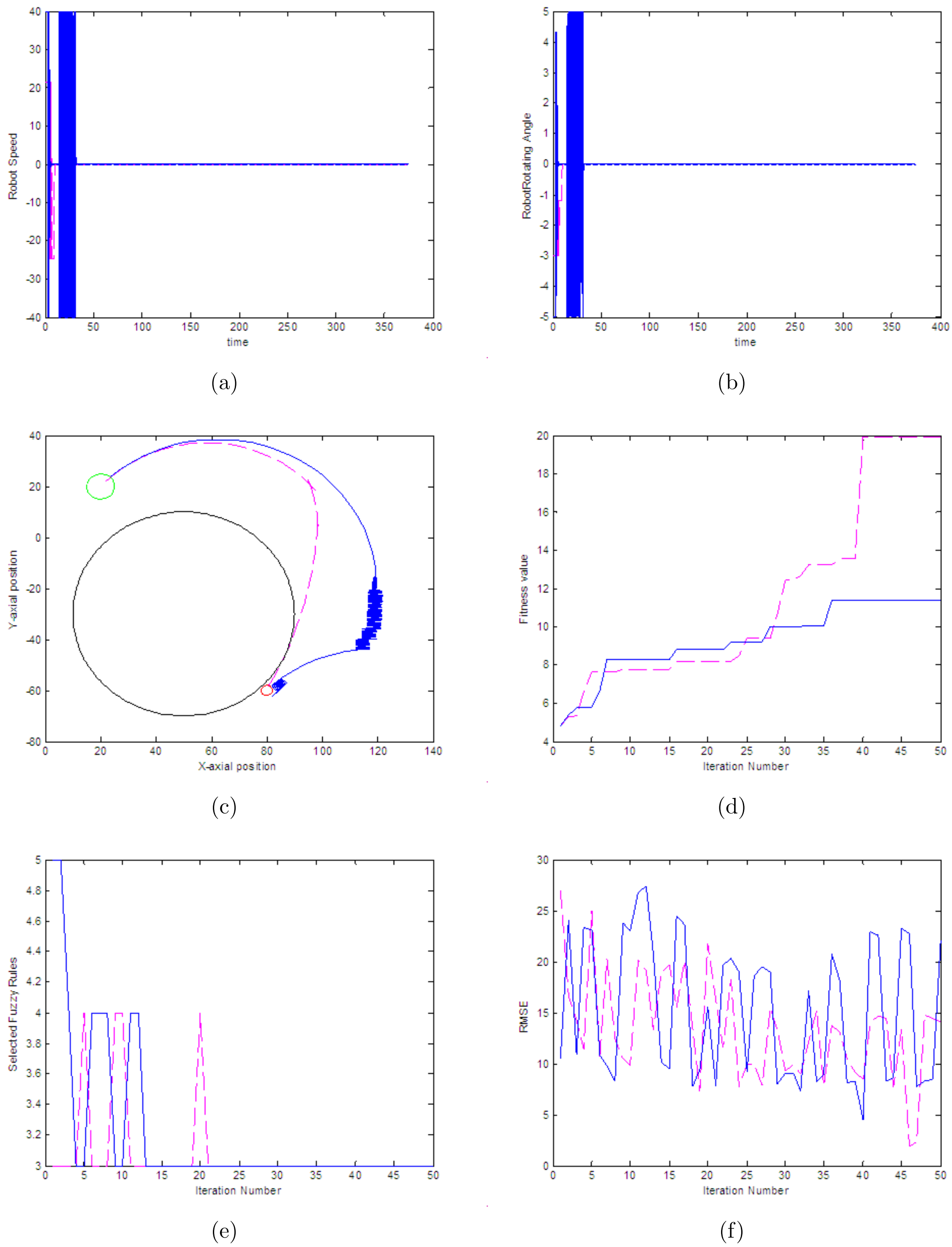

FIGURE 5. Simulation of the first illustrated example. Robot initial position $(20, 20)$, target position $(80, -60)$, block position $(50, -30)$ and block size $= 40$. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) Selected rules number in the respective iteration number and (f) RNSE response for the related iteration number.

Table 1. Performance comparison in this illustrated six robot problems

Performance \ Example		**Different Block Sizes**			**Various Initial and Block Sizes**		**Two Blocks**
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
APSO	RMSE	2.0187	2.5393	1.9113	3.9961	1.8020	4.3660
	NUM	3	4	3	3	3	3
RPSO	RMSE	5.9234	5.3299	4.4912	3.9961	2.2060	5.8986
	NUM	3	4	3	3	3	3

angle for the controlled platform by the self-generated fuzzy model/control mobile robot system were sequentially plotted and are shown in Figures 6(a) and 6(b), respectively. As shown in the plot of the path tracking response of Figure 6(c), the two movable mobile robot paths show that it can smoothly and quickly pass through the dangerous obstacle and approximate to the desired point when the resetting block size is 30. The optimal fitness value through the iteration number is shown in Figure 6(d), which was increased as much as possible in the training cycle. The dynamic response in the related iteration for the extracted fuzzy rules number and RMSE are shown in Figures 6(e) and 6(f), respectively. Only three fuzzy control rules were required to successfully support the mobile robot control task after 50 training cycles were completed. The performance comparison showed that the APSO-based learning algorithm achieves superior RMSE results.

In Case 5, the initial robot position started from $(-20, -50)$ *to the target position* $(80, 50)$*, and the location of the block was selected as* $(45, 0)$*. The time response of the line speed and the rotating angle for the controlled mobile robot platform were sequentially plotted and shown in Figures 7(a) and 7(b), respectively. The traveling paths displayed in Figure 7(c) show that the movable fuzzy system generated by the APSO learning algorithm presents a superior solution compared with the RPSO method to lead the mobile robot driver through the block area and gradually move to the desired target. The optimal fitness values are shown in Figure 7(d), which were increased as high as possible with respect to the training cycle. The simulations shown in Figure 7(e) demonstrated that three appropriate fuzzy rules were sufficient to manage the nonlinear mobile robot movement. The smaller RMSE selected by the learning algorithm is shown in Figure 7(f). These experiments show that the APSO-based fuzzy model/control system can efficiently rebuild the required behavior of the mobile robot and obtain the proper control signal to guide the robot platform to the desired target with a smooth tracking path.*

Example 4.3. ***Cases with two blocks of different sizes:*** *This experiment used two blocks of different sizes. Block 1 was assigned at the* $(0, -40)$ *position, and its block size = 10. Block 2 was located at position* $(20, -40)$*, and its block size = 15. The robot platform started from an initial* $(-20, 40)$ *position to the target* $(-20, -80)$*. Ten maximum rules were used to generate the appropriate fuzzy rules in this case study. The fuzzy model/control system contained two-input and three-output variables, totaling 90 parameters in 30 swarm particles. Because of the escaping problems of two blocks, the two fuzzy system structures, totaling 180 parameters set* $R_j^* = \{\gamma 1_j, \gamma 2_j, c1_{ij1}, c2_{ij2}, c3_{ij}, c4_{ij1}, c5_{ij2}, c6_{ij}, \delta 1_{ij}, \delta 2_{ij}, \delta 3_{ij}, \delta 4_{ij}, \delta 5_{ij}, \delta 6_{ij}, y1_{ij}, y2_{ij}, y3_{ij}, y4_{ij}; 1 \le i \le 10; 1 \le j \le 30\}$*, were efficiently extracted by the proposed APSO and RPSO algorithms simultaneously. Based on the software simulation results, the time responses of the line speed and the turning angle for the mobile robot are shown in Figures 8(a) and 8(b), respectively. The tracing path of the movable mobile robot shown in Figure 8(c) demonstrates that the robot can pass through two obstacles toward the desired point using the appropriate fuzzy control rules. The optimal fitness value against the iteration number is shown in Figure 8(d). The trace*

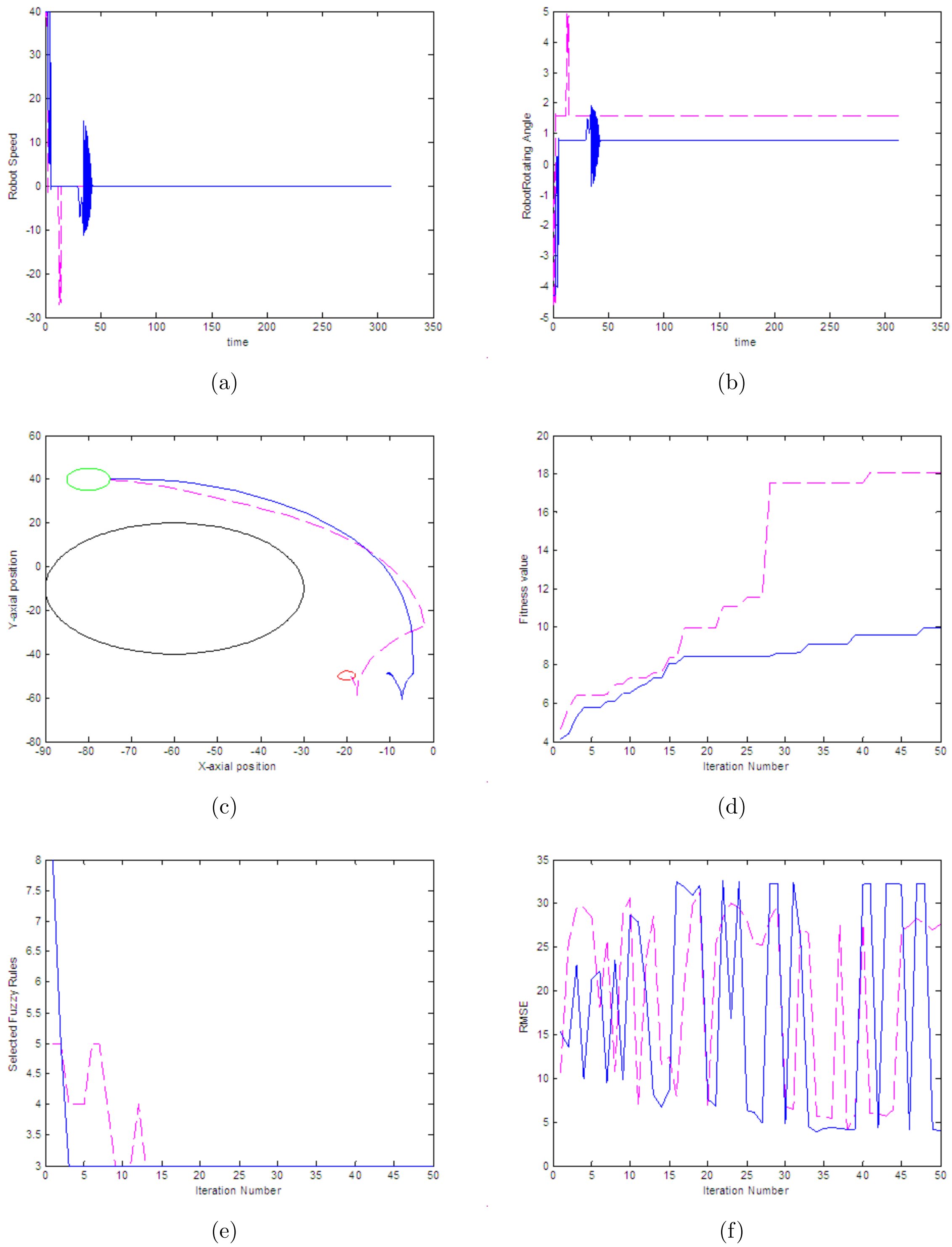

FIGURE 6. Simulation of the second illustrated example. Robot initial position $(-80, 40)$, target position $(-20, -50)$, block position $(-60, -10)$ and block size $= 30$. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) selected rules number in the respective iteration number and (f) RNSE response for the related iteration number.

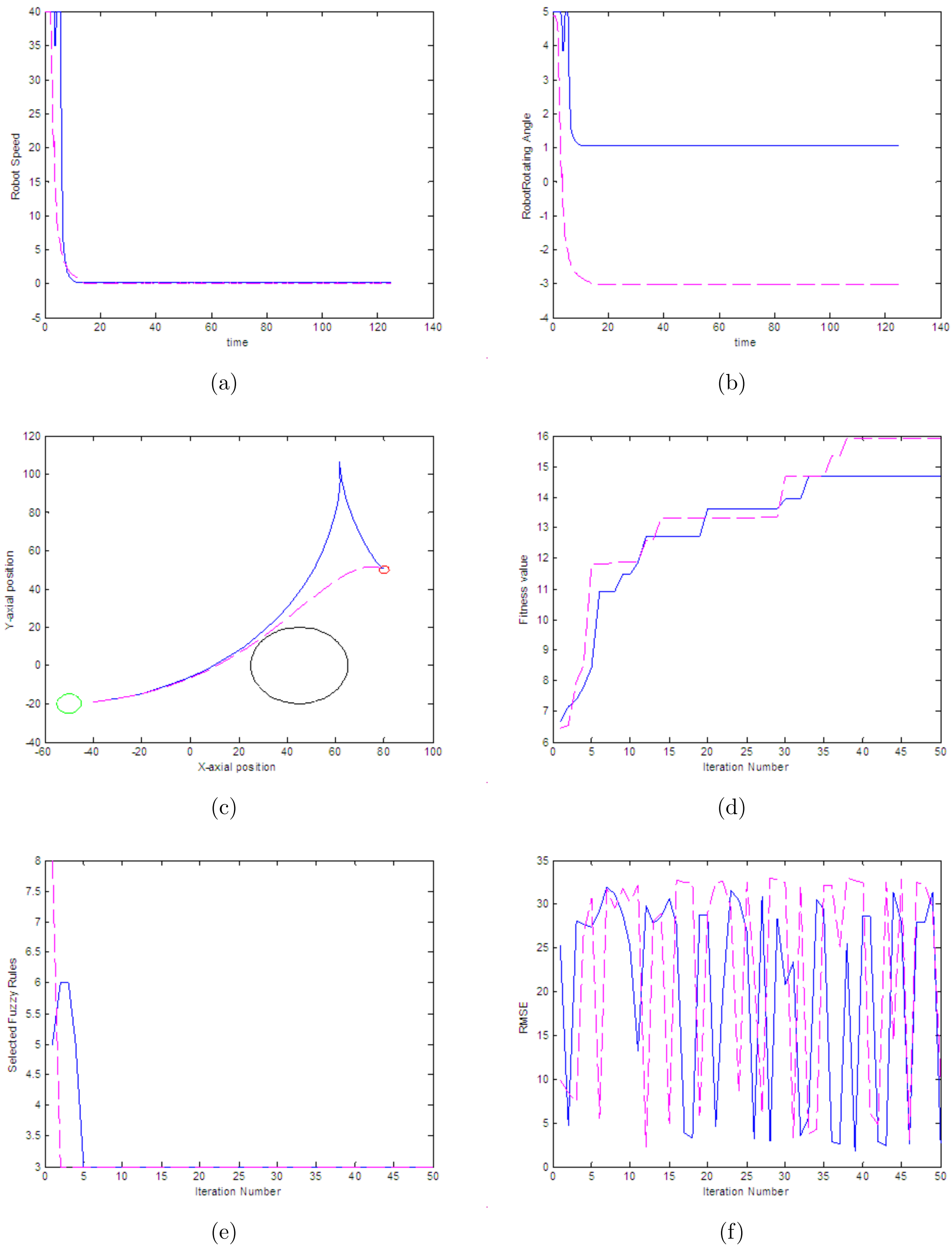

Figure 7. Simulation in different initial condition. Robot initial position $(-20, -50)$, target position $(80, 50)$, block position $(45, 0)$ and block size $= 20$. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) selected rules number in the respective iteration number and (f) RNSE time response for the related iteration number.

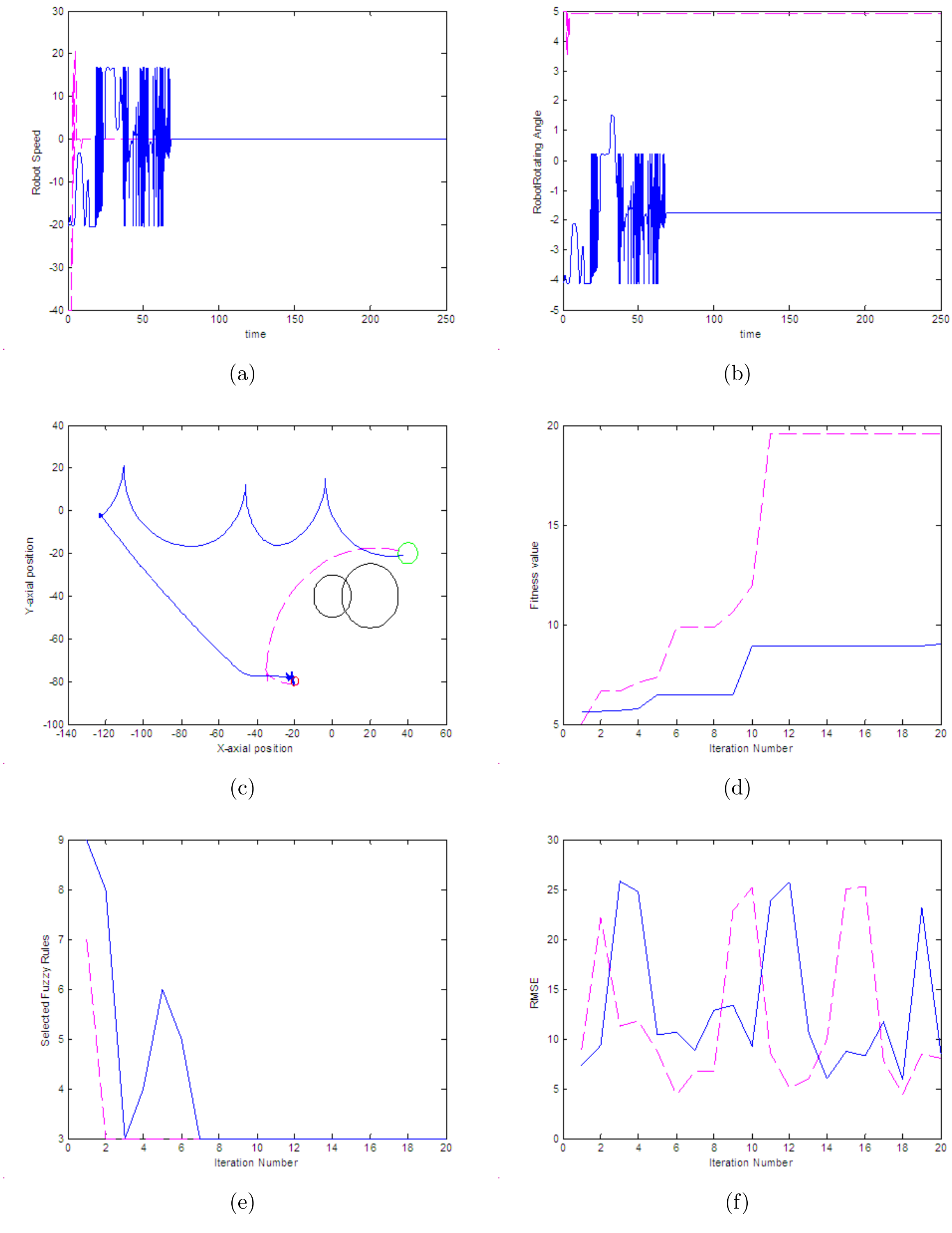

FIGURE 8. The illustrated two blocks example. Robot initial position $(-20, 40)$, target position $(-20, -80)$, block 1 position $(0, -40)$, its block 1 size $= 10$; block 2 position $(20, -40)$ and its block 2 size $= 15$. (a) Time response for the line speed, (b) time response for the rotating angle, (c) tracing representation in the X-axial and Y-axial space, (d) fitness-value against the iteration number, (e) selected rules number in the respective iteration number and (f) RNSE response for the related iteration number.

of the fitness value was gradually increased to achieve superior fuzzy rule solutions after the training cycle was completed. The simulation response of the selected rules number and RMSE are shown in Figures 8(e) and 8(f), respectively. The three proper fuzzy rules were selected by the APSO learning algorithm to successfully execute the suitable command for controlling the movement of the mobile robot. A performance comparison showed that the APSO-based learning scheme obtains the superior RMSE result. The traveling path in these simulations shows that the APSO learning algorithm can yield a superior traveling path to approach the desired target.

5. **Conclusion.** This study proposes a set of hyper-ellipsoids to define the fuzzy partitions of the fuzzy model/control system. The powerful omnidirectional image processing was manipulated to capture the image features and collect the mathematical image data from the surrounding image patterns of the mobile robot to form the fuzzy structure. Based on the approach of the specific fitness function, the APSO and RPSO leaning algorithms were used to configure the initial structure of the fuzzy model/control system, which exhibited excellent performance and extracted the required number of appropriate fuzzy rules. The evolutional PSO leaning algorithm can simultaneously determine the fuzzy rules number and tune the premise and consequent parameters of fuzzy model/control systems to guide the nonlinear mobile robot platform to the desired target in the complex trace. Three types of collision detection and traveling path tracking problems were applied to illustrate the efficiency of the evolutional APSO and RPSO learning methods. These illustrated examples demonstrated that only small swarm sizes with small fuzzy rules numbers are sufficient to move the mobile robot to the desired target. The constructed image processing-based fuzzy model/control system performed the omnidirectional vision-based procedure for wide detection of the surrounding objects in the environments. The fuzzy model/control system was automatically generated with the evolutionary APSO and RPAO learning method to avoid contact with the obstacles and efficiently reach the desired targets within various nonlinear traveling environments. Performance comparisons from the evaluated simulation results show that the proposed APSO learning algorithm is superior to the RPSO method and can achieve the multiple objectives of shortest traveling length, minimal fuzzy rules, and avoiding contact with the obstacles. The proposed self-tuning APSO algorithm is an efficient method to automatically accomplish the complicated design of the model/control fuzzy system without the specific domain knowledge. The omnidirectional image sensor-based mobile robot model is also visible in the future applications to implement the hardware design without choosing the position and direction of sensor.

Acknowledgment. This research was partly supported by the National Science Council of Taiwan under contract NSC 96-2221-E-507-004 & NSC 99-2221-E-507-004.

REFERENCES

[1] F. A. Jafar, Y. Suzuki, Y. Tateno, K. Yokota and T. Matsuoka, An environmental visual features based navigation method for autonomous mobile robots, *International Journal of Innovative Computing, Information and Control*, vol.7, no.3, pp.1341-1355, 2011.
[2] C.-Y. Chen and H.-M. Feng, Hybrid intelligent vision-based car-like vehicle backing systems design, *Expert Systems with Applications*, vol.36, no.4, pp.7500-7509, 2009.
[3] A. Chatterjee and F. Matsuno, A geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots, *Expert Systems with Applications*, vol.37, no.8, pp.5542-5548, 2010.
[4] H.-M. Feng, Self-generation fuzzy modeling systems through hierarchical recursive-based particle swarm optimization, *Cybernetics and Systems: An International Journal*, vol.36, no.6, pp.623-639, 2005.

[5] H.-M. Feng, Self-generation RBFNs using evolutional PSO learning, *Neurocomputing*, vol.70, no.1-3, pp.241-251, 2006.
[6] H.-M. Feng, Hybrid stages particle swarm optimization learning fuzzy modeling systems design, *Tamkang Journal of Science and Engineering*, vol.9, no.2, pp.167-176, 2006.
[7] H.-M. Feng, C.-Y. Chen and J.-H. Horng, Intelligent omni-directional vision-based mobile robot fuzzy systems design and implementation, *Expert Systems with Applications*, vol.37, no.5, pp.4009-4119, 2010.
[8] M. Fiala and A. Basu, Robot navigation using panoramic tracking, *Pattern Recognition*, vol.37, no.11, pp.2195-2215, 2004.
[9] L. Freda and G. Oriolo, Vision-based interception of a moving target with a nonholonomic mobile robot, *Robotics and Autonomous Systems*, vol.56, no.6, pp.419-432, 2007.
[10] R. C. Gonzalez, R. E. Wood and S. L. Eddins, *Digital Image Processing Using Matlab*, Prentice Hall, 2004.
[11] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proc. of IEEE Int. Conf. Neural Networks*, Perth, Australia, pp.1942-1948, 1995.
[12] J. Kennedy, The particle swarm: Social adaptation of knowledge, *Proc. of Int. Conf. Evolutionary Computation*, Indianapolis, IN, USA, pp.303-308, 1997.
[13] J. B. Mbede, P. Ele, C.-M. Mveh-Abia, Y. Toure, V. Graefe and S. Ma, Intelligent mobile manipulator navigation using adaptive neuro-fuzzy systems, *Information Sciences*, vol.171, no.4, pp.47-474, 2005.
[14] K. Samsudin, F. A. Ahmad and S. Mashohor, A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot, *Applied Soft Computing*, vol.11, no.2, pp.1631-1637, 2011.
[15] M. Wang and J. N. K. Liu, Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, *Robotics and Autonomous Systems*, vol.56, no.7, pp.625-643, 2008.
[16] R. L. Williams II, B. E. Carter, P. Gallina and G. Rosati, Dynamic model with slip for wheeled omni-directional robots, *IEEE Transactions on Robotics and Automation*, vol.18, no.3, pp.285-293, 2002.
[17] C.-C. Wong and C.-C. Chen, A hybrid clustering and gradient descent approach for fuzzy modeling, *IEEE Trans. Systems, Man and Cybernetics*, vol.29, pp.686-693, 1999.
[18] C.-C. Wong and C.-C. Chen, A GA-based method for constructing fuzzy systems directly from numerical data, *IEEE Trans. Systems, Man and Cybernetics*, vol.30, no.6, pp.905-911, 2000.
[19] L. A. Zadeh, Fuzzy sets, *Inform. Control*, vol.8, pp.338-353, 1965.