

Design of Path Planning and Obstacle Avoidance for a Wheeled Mobile Robot

Wei-Jen Chen¹ · Bing-Gang Jhong¹ · Mei-Yung Chen¹

Received: 1 February 2016/Revised: 8 June 2016/Accepted: 29 June 2016/Published online: 8 August 2016
© Taiwan Fuzzy Systems Association and Springer-Verlag Berlin Heidelberg 2016

Abstract This paper proposed a path tracking method for wheeled mobile robot (WMR). This method includes path planning and controller design. In path planning, using B-spline instead of A* algorithm to create smooth and obstacle-avoidance path, so that the possibility of collision can be reduced statistically. In controller design, genetic algorithm (GA) and fuzzy logic control (FLC) are combined together for the velocity control for WMR, which is supposed to move in a certain environment. The suitable membership function of FLC according to the feature of environment, such as distance and angle of WMR, is based on the B-spline path, and the input and output of FLC can be adjusted by GA. Therefore, the WMR can move along the planned path steady, which means our method has good performance.

Keywords Mobile robot · A* algorithm · Genetic algorithm · Fuzzy logic control

1 Introduction

Recently, robots are paid more and more attention by many people [1–3]. According to the structure of mobile robot, we can classify into foot robots and wheeled robots. The former, with complex mechanism design problem, should be considered balancing themselves during working. Literatures [4–6] have discussed these issues. Conversely, the latter are simplified as mechanism, we only have to know each part of robot's size, such as platform, wheels, and

sensors, then deriving mathematical model from their structure and consider how to control them [7–9].

In past research, one of the most important task of WMR is path planning, which is to navigate the robot to the destination quickly and safely. In order to satisfy these demands, especially for obstacle avoidance [10], the location of robot is an indispensable information for path planning. If the robot moves in the unknown surrounding, the technology, such as simultaneous localization and mapping, can be used helpfully with the information from camera or ultrasonic range sensor. However, this paper focuses only on the known surrounding to reduce the complexity of system.

In order to achieve the shortest path and the lowest time cost, we adopt A* algorithm to find the shortest path [11]. Comparing with the well-known Dijkstra's algorithm [12], the latter can also be guaranteed to find the shortest path, but its performance is not as fast as the former. When Dijkstra's algorithm does not likely seem to be applied to some situations considering time cost, the improved algorithm of Dijkstra, A* algorithm, is regarded as the more suitable method in this paper.

The idea of B-spline curve was proposed by Schoenberg in 1946 [13]. Others like Bezier curve [14] or Cubic curve [15] are used to design obstacle-avoidance paths for robots. However, B-spline curve is more widely used than the others for its ability of local controlling, adding control points of curves without changing their orders, and bringing up different order curves when fixing B-spline's control point on the other hand.

Genetic algorithm, one of the evolutionary algorithms, is a search heuristic algorithm [16] to solve optimization in the artificial intelligence field of computer science. This heuristic is usually used to generate useful solutions for optimization and searching problems. Evolutionary algorithm is originally developed by some phenomena among

✉ Mei-Yung Chen
cmy@ntnu.edu.tw

¹ Department of Mechatronic Engineering, National Taiwan Normal University, Taipei, Taiwan

evolutionary biology, these phenomena include genetic, mutation, natural selection and hybridization.

Genetic algorithm really has been widely used among bioinformatics [17], phylogenetic, computer science [18], engineering, economics, chemistry, manufacturing, mathematics, physics, science, and other areas of drug measurement, such as C wing, air plane, water supply, games, TSP, learning scenes, coding A face, freeze/flood, simulation crime, etc.

Controlling is a new and evolving subject; there were some names such as expert control [3], learning control, and others have been mentioned beforehand. Intelligent control has been more able to be accepted recently, such as fuzzy logic control (FLC) which is a thinking pattern applied using human reasoning to decide how to control models, that is imitating the way people think of fuzzy phenomenon or things, by way of experts' judgment and then make decision to solve complex control problems. Genetic algorithm is based on the theory of evolution, belonging to wise method [19]. Wisdom control is an effective way of system's control, but this does not mean that the traditional control is no wisdom, we may better say that intelligent control uses human intelligence to handle affairs, and make some decisions. On the contrary, traditional control way uses mathematical derivation out of control, and it's less thinking action in the control process.

In modern control, three feature objects are often encountered and they are given in the following:

- *Uncertainty* Model of the controlled substance not available or uncertain, unsure of model parameters.
- *Nonlinear* Nonlinear increasingly complex-controlled body presented, even nonlinear control theory, also not perfect, it would not be able to solve all the nonlinear problems.
- *Complex Objectives* For example, robust control not only considered performance, but considered robust features about foreign interference and internal system's parameters. Mission requirements are becoming increasingly complex, as a result, only using traditional control will not be applied.

GA is a kind of evolutionary algorithms that can deal with optimization problem. Fuzzy controller is a language controller that allows the operator to easily use natural language to access man-machine dialogue, and do not have to establish a complete mathematical model of the controlled object. In the past, many experts have been able to combine the two ways of dealing with the path planning and obstacle-avoidance problems in WMR [3, 20].

All of the above applications are to solve the mathematical maximization or minimization questions. If function is differentiable, it can be optimized by gradient methods; if not, there are other ways, such as simulation

annealing, random search method, etc. There is a common feature among these methods, the results found may be global optimum or local optimum. And the best advantage of genetic algorithm is avoiding falling into local optimum point during the optimization process of seeking, so the chances to find the best global point would be great. This is mainly accredited to the ability of induced mutations. Even if genetic algorithm may not find the best answer every time, the result always appears near the best answer, unlike the other methods with big opportunity that have their answers fall on the second best point.

This paper is organized as follows. In Sect. 2, we establish some mathematical model of WMR used for motion discussion and robotic control. In Sect. 3, description to the algorithms' theorem about A* algorithm and B-spline curve used for path planning is given in this paper. In Sect. 4, the part concept of GA which is used in this paper is selected and described. In Sect. 5, description to the theorem of FLC and flow of control design is given. In Sect. 6, adoption to the simulation of path planning and path tracking way is given.

2 Kinematic of Wheeled Mobile Robot

Wheeled robot used in this study is composed of three rigid bodies, each of them are platform, fixed wheel installed on each sides of the platform.

The purpose of this part is to get the motion equations of the robot, so we need to analyze its structure, and then derive the geometric and kinematic constraints.

In Fig. 1, let body1 be platform, body2 and body3 are the wheels fixed to the robot left and right sides, and the freedom wheel is not be considered in the derivation process.

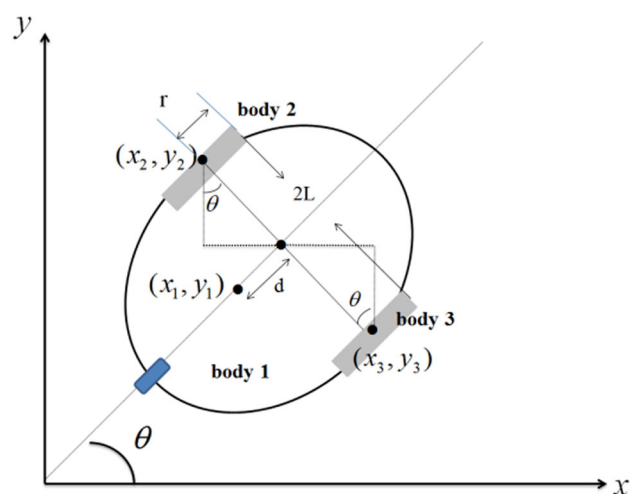


Fig. 1 Define body of WMR

In the following we define some parameters for robot: w_r is radius of the wheel, L is half distance between two wheels. In the three-dimensional space, each of the rigid body needs six variables to describe the relationship between movement and rotation, so in this study we use $x = (x_i, y_i, z_i, \varphi_i, \delta_i, \theta_i)$, $i = 1, 2, 3$, represents variables, (x_i, y_i, z_i) is described as rigid body's coordinates on the inertial coordinate axis, then $(\varphi_i, \delta_i, \theta_i)$ is described as attitude angle of rigid body, each of them represents the roll angle, the pitch angle, and the yaw angle, and i is the corresponding number of rigid mentioned beforehand.

Here define X -axis as the moving direction of robots, Z -axis as direction toward the earth, Y -axis is right-handed coordinate system from X -axis and Z -axis. Therefore, according to this definition we can get $(\varphi_i, \delta_i, \theta_i)$ as the three rotation angles of shaft on X -, Y -, Z -axis.

Because attitude angle is defined in the robot, it is relative with robot's movement, and each wheel has common axis and is perpendicular to the moving direction, therefore the angular direction of the wheel and platform is consistent and then we get:

$$\theta_2 = \theta_1 \quad (1)$$

$$\theta_3 = \theta_1 \quad (2)$$

The position of rigid is affected by robot's structural constraints, therefore there are these following constraints:

$$x_2 = x_1 + d \cos \theta_1 - L \sin \theta_1 \quad (3)$$

$$y_2 = y_1 + d \sin \theta_1 + L \cos \theta_1 \quad (4)$$

$$x_3 = x_1 + d \cos \theta_1 + L \sin \theta_1 \quad (5)$$

$$y_3 = y_1 + d \sin \theta_1 + L \sin \theta_1 \quad (6)$$

After the above six robot motion model geometric constraints, the next step includes motion restrictions and we derive it.

Considering wheels are pure rolling state on the ground, the moving speed is relative to the rotation angular velocity and the robot's moving direction, as per the following four sports constraints:

$$\dot{x}_2 = r \dot{\delta}_2 \cos \theta_1 \quad (7)$$

$$\dot{y}_2 = r \dot{\delta}_2 \sin \theta_1 \quad (8)$$

$$\dot{x}_3 = r \dot{\delta}_3 \cos \theta_1 \quad (9)$$

$$\dot{y}_3 = r \dot{\delta}_3 \sin \theta_1 \quad (10)$$

By (3), (5), we can get:

$$x_3 - x_2 = 2L \sin \theta_1 \quad (11)$$

Differential form of (11) is the following:

$$\dot{x}_3 - \dot{x}_2 = 2L \cos \theta_1 \cdot \dot{\theta}_1 \quad (12)$$

Including motion constraints (7) and (9) we have the following:

$$w_r(\dot{\varphi}_2 - \dot{\varphi}_3) = 2L\dot{\theta}_1 \quad (13)$$

Since this equation can be integral, it can be converted into a geometric constraint:

$$\theta_1 = \frac{r}{2L}(\delta_3 - \delta_2) + c_2, \quad (14)$$

where C_2 is a constant, θ_1 represents initial value. In addition, from (3) to (6) also can be obtained:

$$x_2 + x_3 = 2x_1 + 2d \cos \theta_1, \quad (15)$$

$$y_2 + y_3 = 2y_1 + 2d \sin \theta_1, \quad (16)$$

respectively, and to the left after the equal sign and the differential was:

$$\dot{x}_1 = \frac{1}{2}(\dot{x}_2 + \dot{x}_3) - d \cos \dot{\theta}_1 \cdot \theta_1 \quad (17)$$

$$\dot{y}_1 = \frac{1}{2}(\dot{y}_2 + \dot{y}_3) + d \cos \dot{\theta}_1 \cdot \theta_1 \quad (18)$$

The motion constraints (7) to (10) after the substitution and they were analyzed to obtain:

$$\dot{x}_1 = \frac{w_r}{2}(\dot{\varphi}_2 + \dot{\varphi}_3) \cos \theta_1 - d \cos \dot{\theta}_1 \cdot \theta_1 \quad (19)$$

$$\dot{y}_1 = \frac{w_r}{2}(\dot{\varphi}_2 + \dot{\varphi}_3) \sin \theta_1 + d \cos \dot{\theta}_1 \cdot \theta_1 \quad (20)$$

Compared with the platform coordinates (x_1, y_1) , the central reference point $Pr(x_r, y_r)$ around the axle use it more often, and because the wheel is mounted on the servo motor, it can be controlled directly, in this order $\dot{\delta}_2 = u_1$ and $\dot{\delta}_3 = u_2$, and finally we can get sports model robots:

$$\dot{\delta}_2 = u_1$$

$$\dot{\delta}_3 = u_2$$

$$\dot{x}_r = \frac{r}{2}(u_1 + u_2) \cos \theta_1 \quad (21)$$

$$\dot{y}_r = \frac{r}{2}(u_1 + u_2) \sin \theta_1$$

$$\dot{\theta}_1 = \frac{r}{2L}(u_1 - u_2)$$

3 Path Planning

3.1 A* Algorithm

A* algorithm determines the lowest path-limited nodes by least cost [21]; mathematic form is represented as (22). This algorithm contains two stages: search and reconstruction phase. When $g(\mathbf{n})$ represents the actual distance from the start point n_0 to any node n , and $h(\mathbf{n})$ represents any node n to estimate the distance to the target point n_g ,

the overall cost can be expressed as a function of the distance

$$f(\mathbf{n}) = g(\mathbf{n}) + h(\mathbf{n}) \quad (22)$$

When $h(\mathbf{n})$ is less than the actual distance from the node n to the destination point, then you can certainly get the shortest path, and if $h(\mathbf{n})$ is close to the practical cost, this algorithm will be the most efficient. But if $h(\mathbf{n})$ is larger than the practical cost, the result will not be guaranteed. In the practical operation, we usually define $h(\mathbf{n})$ to be the distance between current node and target point. Operation process of A* algorithm is as follows and the result is as shown in Fig. 2.

Step 1 to Step 10 belong to searching stage, it's to find the goal point with the lowest cost, and Step 11 to Step 13 belong to reconstruction stage, it's to establish the result of path with all the nodes that had been found.

- Step 1 Establish *OPEN* array and *CLOSE* array. *OPEN* array sets forthcoming nodes that will be estimated, *CLOSE* array sets the nodes which was completed computing. *OPEN* array as the starting point n_0 ; *CLOSE* array was empty. $g(\mathbf{n})$ as the actual lowest cost from start point n_0 to the node n , and it is possible to obtain $g(n_0)$ as zero. In this step while estimating $h(n_0)$, the result $g(n_0) = 0$ to obtain $f(n_0) = h(n_0)$.
- Step 2 Construct a loop from step 3 to step 10, if *OPEN* array is nonempty, then calculate into the loop, it means it still did not arrive to the end, and after judgment, if no path to the end is generated, then return fail plan.
- Step 3 From *OPEN* array find the node which has smallest $f(n)$, and regard it as next current node x , determining whether it is the end point n_g , if so, it means that the search step is completed, then remove the target node from *OPEN* array and add to *CLOSE* array, out of step 2 and go to step 11.
- Step 4 Search in the vicinity from the current node and reachable nodes defined as y_i .

- Step 5 Construct a loop containing step 5 to step 10, judging each nodes y_i in the loop until all adjacent node judgment is completed.
- Step 6 Determine whether the nodes y_i are in the *CLOSE* array, skipping this node, determining the next adjacent nodes, or continue to the next step.
- Step 7 Evaluating $g(y_i)$ of y_i , this function is defined by the current actual cost function node x , and can be expressed as $\hat{g}(y_i) = g(x) + \text{dis}(x, y_i)$, which $\text{dis}()$ function represents linear distance between nodes.
- Step 8 Setting a Boolean parameter k is false, recording whether the node information needs to be updated.
- Step 9 Determining whether nodes y_i are in *OPEN* array, if not, the node y_i will be added in *OPEN* array, and define whether parameter k is true, or it is again judged $\hat{g}(y_i)$ whether it is less than $g(y_i)$ at node y_i , if so, parameter k is then set true, to indicate that the node data need to be added or modified.
- Step 10 Analyzing the parameters k , if it is *false*, then go straight back to Step 5, if it is *true* then update the information, comprising: set node y_i source node as x , it is represented by the current node x and $g(y_i)$ substituted by $\hat{g}(y_i)$ indicating that there is a lower cost path.
- Step 11 This is the reconstruction stage, let the goal point n_g of the path temporarily store P first position, i.e., $P(1) = n_g$.
- Step 12 Constructing of a loop, in accordance with $P(i + 1) = c(P(i))$, filling N nodes information gradually in the temporarily stored path, until the starting point is added to the temporarily stored path, which satisfies $P(N) = n_0$.
- Step 13 Overturning the order in temporarily stored path, namely $P(i) = P(N - i + 1)$, $i = 1, 2, \dots, N - 1$, then will have the correct order path P , namely to complete the A* algorithm.

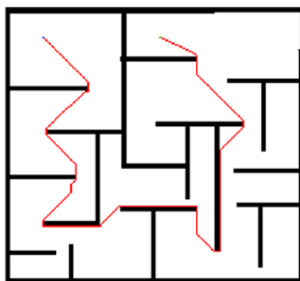


Fig. 2 Path planning by A* algorithm

3.2 B-Spline Curve

Path planning is not usually able to accurately design a path to meet user's demand by one type of algorithm; the results from the previous design way in Fig. 2 shows that there are too many corners during the path that are not continuous and not suitable for real mobile robots to track with their body size, in order to overcome such problems, having characteristics of local adjusting with B-Spline curve will look quite apply.

B-spline curve is generated by a set of base functions and parameters of control points, consisting of affine invariant, adjustments in the local curve, convex hull property, continuity and differentiability superior qualities, etc., making it more suitable as a smooth description method curve. *B-spline* curve's mathematical model is represented as follows:

$$P(t) = \sum_{i=0}^n P_i N_{i,k}(u), \tag{23}$$

where $P(t)$ is position vector on the curve along with parameter t , B_i is position vector control points, also named control points, the number of it is $n + 1$, $N_{i,k}$ is normalized *B-spline* basis function. At the order k (degree $k - 1$), Cox-de Boor regression formula of i_{th} basis function $N_{i,k}(t)$ is defined as follows:

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } t_i \leq u < t_{i+1} \\ 0 & \text{others} \end{cases} \tag{24}$$

and

$$N_{i,k}(u) = \frac{(u - t_i)N_{i,k-1}(t)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u)N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \tag{25}$$

defines $0/0 = 0$ during calculating.

t_i is named as knot value, and $\{t_i\}$ is named as knot value, and the knot value is defined as (26)

$$\begin{aligned} t_i &= 0, & i &= 0, \dots, k - 1 \\ t_i &= i - k + 1, & i &= k, \dots, n \\ t_i &= n - k + 2, & i &= n + 1, \dots, n + k \end{aligned} \tag{26}$$

x_i is the i th element of node vector, and $x_i \leq x_{i+1}$, t is between t_{\min} and t_{\max} . In (25). For $k = 3$ as example, assuming $k = 3$, we then used (23)–(26) and then the basic function is given in Table 1 and the result is shown in Fig. 3. $k = 3$ (order = 3), k not value: $\{0, 0, 0, 1, 2, 3, 3, 3\} = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ by the procedure and by defining the position of control point we can find the value $P(t)$.

The curve of order k (degree $k - 1$) usually satisfies the following conditions:

- $P(t)$ should be the polynomial degree $k - 1$ in any region of x_i .
- $P(t)$ contains both differential and continuity of 1 to $k - 2$ order.
- In the valid region of t , $N_{i,k} \geq 0$, and each of $N_{i,k}$ has only one maximum value, and $N_{i,k}$ is the continuous smooth curve except $k = 1, 2$.

Table 1 Basic function of $k = 3$

$N_{0,1} = \begin{cases} 1, & u = 0 \\ 0, & \text{others} \end{cases}$	$N_{0,3} = \frac{u - t_0}{t_2 - t_0}, \quad N_{0,2}(u) + \frac{t_3 - u}{t_3 - t_1}$
$N_{1,1} = \begin{cases} 1, & u = 0 \\ 0, & \text{others} \end{cases}$	$N_{1,2}(u) = (t_3 - u), \quad N_{1,2} = \begin{cases} (1 - u)^2, & 0 \leq u \leq 1 \\ 0, & \text{others} \end{cases}$
$N_{2,1} = \begin{cases} 1, & 0 \leq u \leq 1 \\ 0, & \text{others} \end{cases}$	$N_{1,3} = \begin{cases} \frac{u(2 - u)}{2} + u(1 - u), & 0 \leq u \leq 1 \\ \frac{(2 - u)^2}{2}, & 1 \leq u \leq 2 \\ 0, & \text{others} \end{cases}$
$N_{3,1} = \begin{cases} 1, & 1 \leq u \leq 2 \\ 0, & \text{others} \end{cases}$	$N_{2,3} = \begin{cases} \frac{u^2}{2}, & 0 \leq u \leq 1 \\ \frac{u(2 - u) + (3 - u)(u - 1)}{2}, & 1 \leq u \leq 2 \\ \frac{(3 - u)^2}{2}, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$
$N_{4,1} = \begin{cases} 1, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$	$N_{3,3} = \begin{cases} \frac{(u - 1)^2}{2}, & 1 \leq u \leq 2 \\ \frac{(u - 1)(3 - u)}{2} + (3 - u)(u - 2), & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$
$N_{0,2} = 0, \quad N_{1,2} = \frac{u - t_1}{t_2 - t_1}$	$N_{4,3} = \begin{cases} (u - 2)^2, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$
$N_{1,1}(u) + \frac{t_3 - u}{t_3 - t_2}, \quad N_{2,1}(u) = \begin{cases} 1 - u, & 0 \leq u \leq 1 \\ 0, & \text{others} \end{cases}$	
$N_{2,2} = \begin{cases} u, & 1 \leq u \leq 2 \\ 0, & \text{others} \end{cases} + \begin{cases} 2 - u, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases} = \begin{cases} u - 1, & 0 \leq u \leq 1 \\ 2 - u, & 1 \leq u \leq 2 \\ 0, & \text{others} \end{cases}$	
$N_{3,2} = \begin{cases} u - 1, & 1 \leq u \leq 2 \\ 3 - u, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$	
$N_{4,2} = \begin{cases} u - 2, & 2 \leq u \leq 3 \\ 0, & \text{others} \end{cases}$	

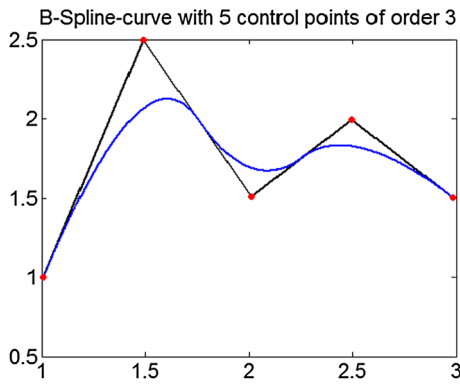


Fig. 3 B-spline curve($n = 4, k = 3$)

4 Genetic Algorithm

Genetic algorithm is usually implemented as a way of computer simulation [21]. It contains Gene, Chromosome, Fitness function, and Population for the elements for algorithm. For an optimization problem, a number of candidate solutions' abstract representation of the population is to evolve to a better solution. Traditionally, the solution is usually represented with binary (strings 0 and 1), but also can use other representation [22]. Evolving begins from completely random individual populations, then occurred on next generations. In each generation, the fitness of the entire population is evaluated, choosing multiple individuals (based on their fitness) randomly from the current population, generating new life populations through natural selection, and the population in the last iteration of the algorithm becomes the current population.

The strategy of genetic algorithm is shown in Fig. 4, its procedure contains initially, evaluation, selection, crossover, mutation, and brings new chromosomes, next we focus on each procedure and discuss them in detail.

- **Encoding.** In order to facilitate mating with crossover and mutation, genes should be encoded into binary digits and called as chromosome. Genetic algorithm which consisted of coded procedures is binary genetic algorithm, if genetic algorithm's parameter is directly composed of floating, it is called real genetic algorithms, continuous parameter genetic algorithms, or floating point genetic algorithms (RGA, CGA, or FPGA). Generally, real number converts to binary number using direct binary conversion (straight binary), for example, if x is from 0 to 10, and represented by bit length number 7, then the decimal and binary number is represented as follows:
 $0 \rightarrow 000\ 0000 (=0)$
 $100 \rightarrow 111\ 1111(=127)$
 then $50 = \text{fix}(0.5 \cdot 127) = 63$ is selected, that is 0111 111. But the number 011 1111 represents $100 \times (63/$

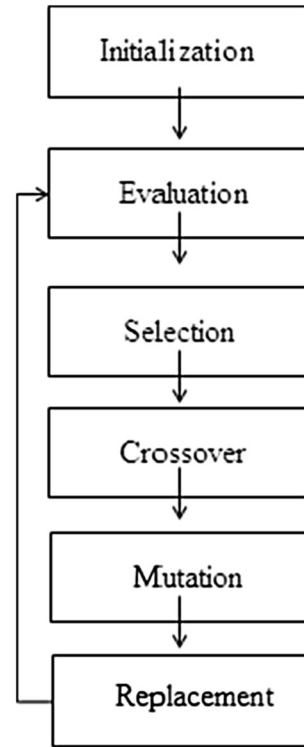


Fig. 4 Procedure of GA

127) = 49.606 actually, rather than 50. Because there are 128 binary numbers, therefore a real number from 0 to 100 can only correspond to 128 bits, instead of infinite number. This is called Quantization. The greater the number of bit length to represent, the more it means more accurate (resolution). The disadvantage is that the longer transcoding time will slow down execution time of program. The conversion formula is

$$x = \text{B2D} \times (\text{UB} - \text{LB}) / (2^l - 1) + \text{LB}, \tag{27}$$

where B2D is transfer from binary to decimal, i.e., $1101 \rightarrow 13$, UB is maximum value, LB is minimum value, L is bit length, x is gene's real value, for example, $\text{UB} = 100$, $\text{LB} = 0$, binary = 0111 1110, then $\text{B2D} = 126$, $L = 8$,

$$x = 126 \times (100 - 0) / (2^8 - 1) + 0 = 49.412$$

Next, for calculating fitness function, we have to transfer binary after encoding procedure to decimal, called "Decoding."

- **Evaluation and Selection** Here we introduce the way Roulette Wheel Selection. In Fig. 3, the proportion of the round can be thought as each selection's probability represented in (28).

$$\begin{cases} f_1 = 4 \\ f_2 = 3 \\ f_3 = 3 \\ f_4 = 2 \end{cases} \quad p_i = \frac{f_i}{\sum_j f_j} \quad (28)$$

f_i is value of fitness function, here defined by number of '1' in binary as Figs. 5 and 6

- **Crossover** Select two sets of data for mating to match, then set new chromosome into the next generation of ethnic groups as in Fig. 7.
- **Mutation** Mutation as shown in Fig. 8 is to produce new varieties so that searching for the best values will not be trapped in a local maximum or local minimum value.

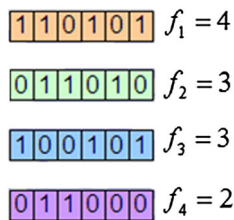


Fig. 5 Value of fitness function

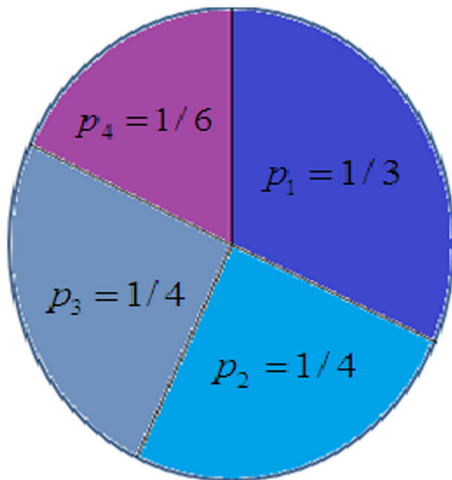


Fig. 6 Roulette wheel selection

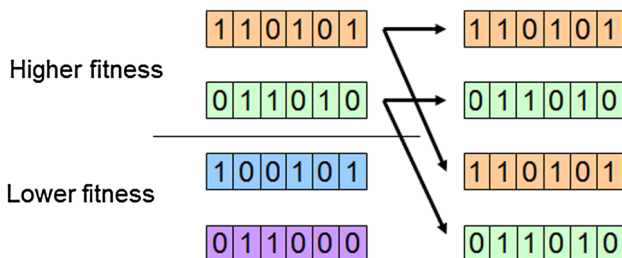


Fig. 7 Crossover in GA

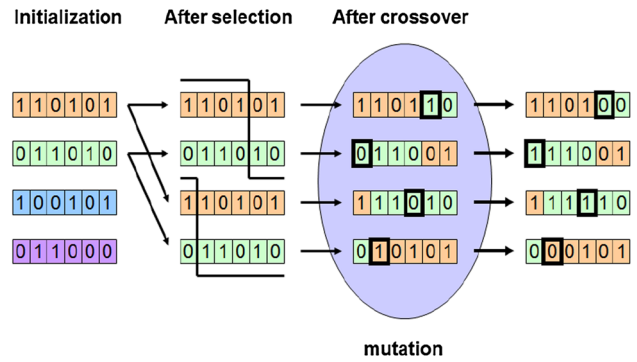


Fig. 8 Mutation in GA

The probability of mutation is signed as, it's value usually provided between 0.01 and 0.08 generally, when set 0 may let genetic algorithms not get the best results, if too large, it can't to be convergence. If there are 100 chromosomes, and $P_m = 0.08$, then the average of eight chromosome will be mutated (determined by the random number generated). General mutation is just select one bit during string randomly, original 0-1, and original 1 becomes zero.

5 Fuzzy Control Theorem

Fuzzy control is a way of using FLC to control plants. FLC is composed according to Fuzzy set theory advocated by Professor Zadeh's in 1965 [23] and consists of an inference to generate control signals. You can design such a controller without a proper mathematical model of the controlled substance (differential equations or transfer functions to start the design). Fuzzy theory now contains fuzzy set, fuzzy logic, fuzzy measure, fuzzy inference system, fuzzy numbers, fuzzy integral, fuzzy entropy, etc.

Fuzzy inference logic does not require complicated mathematical operations (such as differential, integral, or floating point of error), maybe a rule table could complete the entire fuzzy controller inference result. On the other hand, the best benefit of fuzzy controller is saving the excessive computational load during performing table look-up process. It makes the system capable to execute control algorithms sufficiently in small sampling time, and ensures robots' control quality and results of path tracking, Fig. 9 is the content of fuzzy control procedure, and the following describes each site of flow of fuzzy logical control.

- **Fuzzification:** It's the step of conversion input data into equivalent language value, quantifying, and setting up membership function.
- **Fuzzy rule:** It's causal relationship between input and output "IF, THEN" design rules, for example,

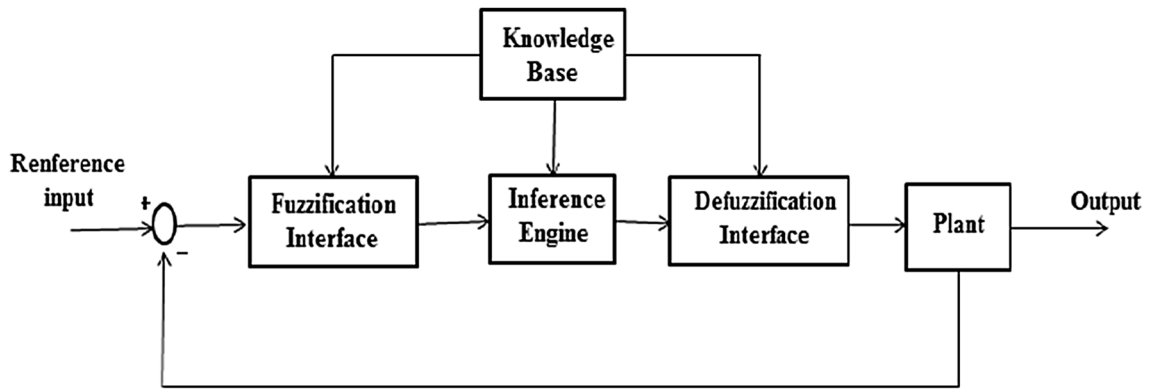


Fig. 9 Procedure of Fuzzy control

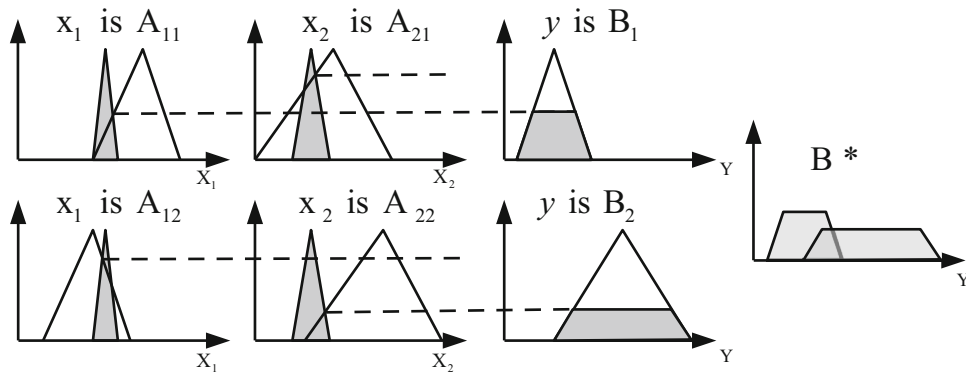


Fig. 10 Max–min operation

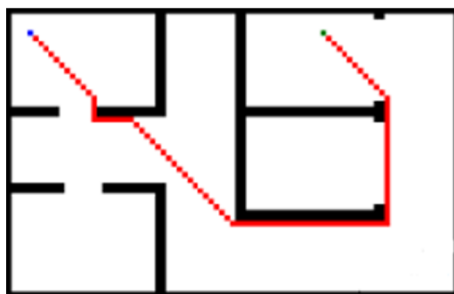


Fig. 11 Path planning of shortest path with A* algorithm

R^1 : IF X_1 is A_{11} AND X_2 is A_{21} THEN Y is B_1
 R^2 : IF X_1 is A_{12} AND X_2 is A_{22} THEN Y is B_2 then
 here we use the “max–min operation” to inference output as shown in Fig. 10.

- Inference Engine: It’s the step that lets fuzzy input set via fuzzy logic inference, combining with all IF–THEN rules then mapped to fuzzy output set via the fuzzy rule base.

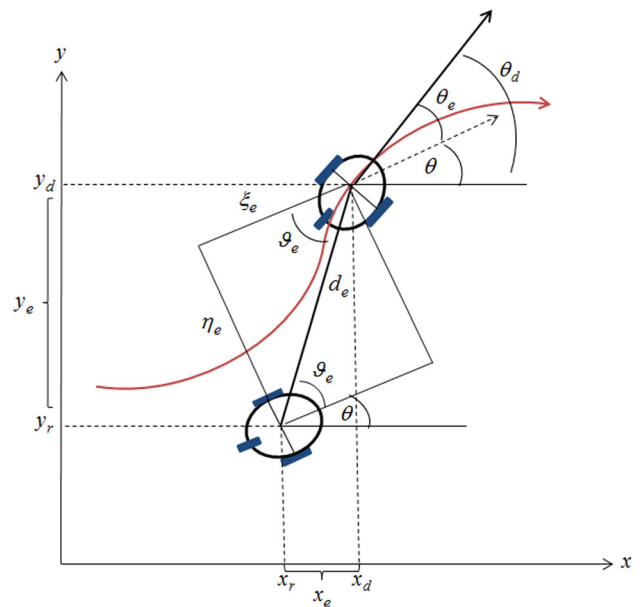


Fig. 12 Tracking of avoidance path

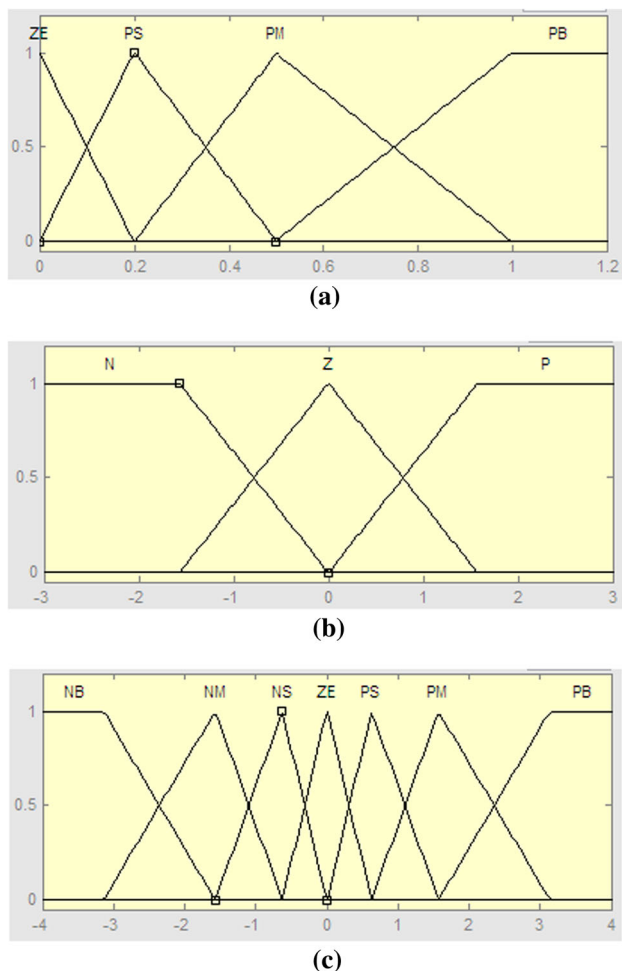


Fig. 13 Membership function of fuzzy input parameter. **a** d_e , **b** θ_e , **c** φ_e

- Defuzzification** The process of converting conclusions generated by the fuzzy inference into explicit value. Here “centroid method” is provided in Fig. 10 as a way of defuzzification to solve overlap region of membership function, define fuzzy set C as conclusion of the fuzzy rules after fuzzy inference procedure, and y is the result of the solution after defuzzification. Following are the two formula forms in different situation.

When the range is continuous

$$y^* = \frac{\int_Y \mu_C(y) \cdot y dy}{\int_Y \mu_C(y) dy}, \tag{29}$$

f represents symbol of continuous.

When the range of discrete time

$$y^* = \frac{\sum_{i=1}^L \mu_C(y_i) \cdot y_i}{\sum_{i=1}^L \mu_C(y_i)}, \tag{30}$$

\sum represents symbol of discrete time. $\mu_C(y_i)$ is i th grade of membership about output’s rule set, and y_i is output value of i th rule.

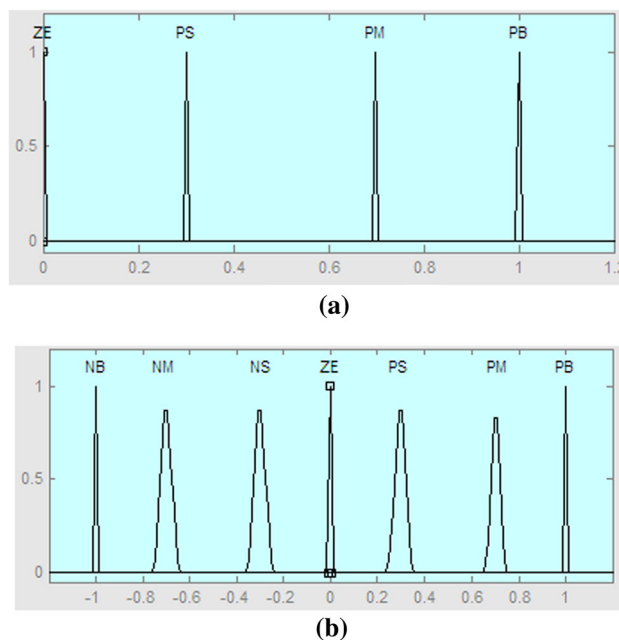


Fig. 14 Membership function of fuzzy output parameter

Table 2 Fuzzy rule of path tracking

θ_e	d_e							
	ZE		PS		PM		PB	
N	Δu_1	Δu_2	Δu_1	Δu_2	Δu_1	Δu_2	Δu_1	Δu_2
Z								
P								

φ_e	d_e							
	ZE	PB	ZE	PM	ZE	PS	ZE	ZE
PB	ZE	PB	ZE	PM	ZE	PS	ZE	ZE
PM	ZE	PM	ZE	PM	PS	PS	PM	ZE
PS	ZE	ZE	ZE	ZE	PM	ZE	PB	ZE
ZE	ZE	ZE	ZE	ZE	PM	ZE	PB	ZE
NS	ZE	NS	ZE	ZE	PM	ZE	PB	ZE
NM	ZE	NM	ZE	NM	PS	NS	PM	ZE
NB	ZE	NB	ZE	NM	ZE	NS	ZE	ZE

6 Simulation

In this section, we try to let the robot move along a path which had been designed, and when robot meets an obstacle, it also can avoid obstacle successfully. We planned the shortest path from start to destination of WMR by A* algorithm and let WMR move along it as shown in Fig. 11; the red line is the shortest path by A* algorithm. Because the real robots should have their specific size, when WMR meets obstacles, we deal with the procedure in avoidance of obstacle by the design of a new obstacle-avoidance path as shown in Fig. 12 [24]. Equation (31) is

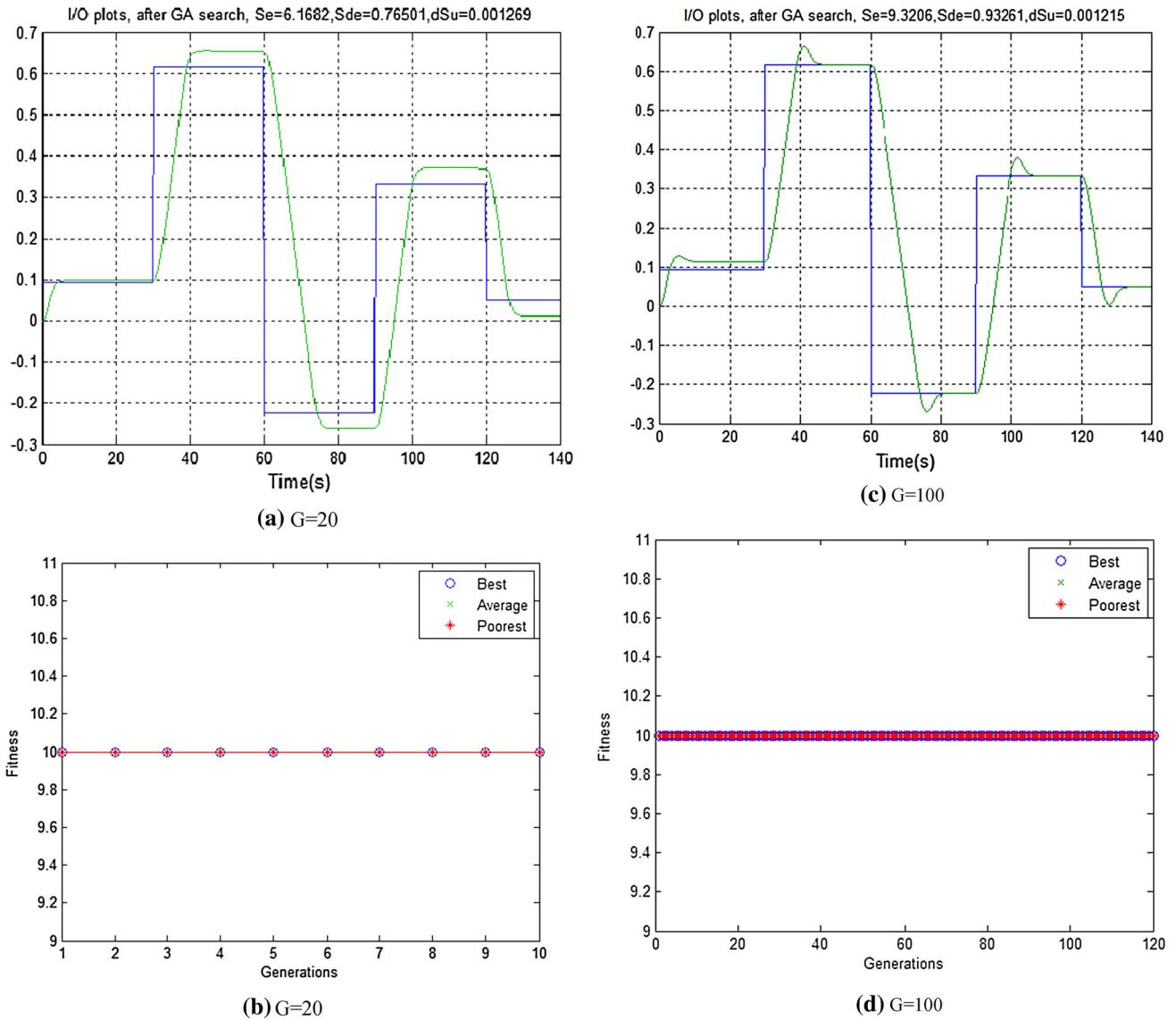


Fig. 15 Velocity control of path tracking (a–d), (a, c) are results of velocity control, (b, d) are the results of evaluation value of GA

transfer from Cartesian coordinate to the new coordinate based on the direction of WMR.

$$\begin{bmatrix} \xi_e \\ \eta_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x_r \\ y_d - y_r \\ \theta_d - \theta_r \end{bmatrix} \quad (31)$$

The error parameters between reference path and WMR’s direction are defined as (x_e, y_e, θ_e) , and

$$x_e = x_d - x_r, y_e = y_d - y_r, \theta_e = \theta_d - \theta_r$$

are also defined as $d_e = \sqrt{\xi_e^2 + \eta_e^2}$, $\vartheta_e = \tan^{-1}(\eta_e/\xi_e)$, d_e is distance between WMR’s current position and goal on the avoidance path, ϑ_e is line of sight angle from WMR’s direction to goal.

After some parameters have been defined, we then use $d_e, \theta_e, \vartheta_e$ as input of FLC, and $\Delta s_1, \Delta s_2$ as output, each $\Delta s_1, \Delta s_2$ is represented as linear and angular velocity of left wheel, respectively. Each of the Figs. 13 and 14 represent the membership function of fuzzy input and fuzzy output. Table 2 shows the suitable fuzzy rule to achieve the proposal of path tracking on new avoidance path planned by B-spline curve, it’s considered by some situation:

- S1 When ϑ_e is close to zero and θ_e is small, it should move fast forward and avoid any turning action, design angular velocity $\Delta s_2 \rightarrow 0$.

- S2 When $d_e \rightarrow 0$, it's much important to adjust Δs_2 than Δs_1 . Here $\Delta s_1 \rightarrow 0$ is defined and Δs_2 is changed with θ_e .
- S3 When d_e is too large, robot will only move in forward motion without rotating ($\Delta s_2 \rightarrow 0$).
- S4 Δs_2 decreases when d_e increases and keeps increasing when $|\vartheta_e|$ and $|\theta_e|$ increase.
- S5 Δs_1 increases when d_e increases and keeps decreasing when $|\vartheta_e|$ and $|\theta_e|$ increase.

If the WMR can avoid obstacles successfully, then it moves along the planned path mentioned before. In the following contents, we consider fuzzy control parameters such as fuzzy rules, fuzzy set, scaling factor, and include chromosome of GA, then after the fitness function is calculated and is compared with each fitness value, we find the optimal fuzzy control to avoid obstacles.

We try to control WMR to move along the avoidance path by velocity control of WMR, and we use GA algorithm to find suitable scaling factor then use different generation numbers (here define G) to achieve the proposal and compare with efficiency between each other as shown in Fig. 15. Fig. 15a, b show the result $G = 20$ and (c, d) are $G = 100$. (a, c) are the velocity results (here they are the design to the right wheel's velocity); (b, d) are the results of evaluate value, blue line is the range of best fitness, red is the range of poorest fitness, and green line is average fitness. In Fig. 15, we can find result such that the more times G is, the better the ideal velocity will be done.

7 Conclusion

In this paper, we planned a suitable path of WMR to move along in an environment containing obstacles. And when WMR finds an obstacle, we planned an avoidance path and let WMR move along it. Here we consider a velocity control way to achieve the proposal using GA algorithm to select the suitable evaluate value as fitness function of the fuzzy control, and by doing so, the result of the path will be shorter and smoother when the WMR meets obstacles.

In future, we will try to add the dynamic obstacles in the environment and use the predicting theorem to predict the next position of the obstacle so that the WMR can avoid them accurately and arrive to the destination rapidly.

References

- Weiteng, Z., Baoming, H., Dewei, L., Bin, Z.: Improved reversely a star path search algorithm based on the comparison in valuation of shared neighbor nodes, *International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 161–164, 2013
- Zhou, H., Chen, W., Fang, K.: A Fuzzy Control algorithm for collecting main pressure controlling using expert control, *International Conference on Intelligent Computation Technology and Automation*, pp.104–107, 2010
- El, S., Su, L.: Fuzzy critical path problem for project network. *Int. J. Pure Appl. Math.* **85**(2), 223–240 (2013)
- Choil, K. C., Lee, H.-J., Lee, M.C.: Fuzzy posture control for biped walking robot based on force sensor for ZMP, *SICE-ICASE International Joint Conference 2006*, pp.1185–1189, 2006
- Park, J.H., Kim, E.S.: Foot and body control of biped robots to walk on irregularly protruded uneven surfaces, *IEEE Transactions on Systems*, vol. 39, no.1, pp. 289–297, 2009
- Suwanratchatamane, K., Matsumo, M., Hashimoto, S.: *IEEE Transactions on Industrial Electronic*, vol. 58, no. 8, 2011
- Ibrahim M. I., Johari J.: Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach, *U.S. Government work not protected by U.S. copyright*
- Liang, X., Wang, H., Chen, W., Guo, D., Liu, T.: Adaptive image-based trajectory tracking control of wheeled mobile robots with an uncalibrated fixed camera, *IEEE Transaction on Control Systems Technology*, vol. 23, no. 6, pp. 2266–2282, 2015
- Linda, O., Manic, M.: Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data, *Transactions on Industrial Electronics*, vol. 58, no. 8, pp 3187–3195, 2011
- Fu, G., Corradi, P., Menciassi, A., and Dario, P.: An integrated triangulation laser scanner for obstacle detection of miniature mobile robots in indoor environment, *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 778–783, 2011
- Oriolo, G., Ulivi, G., Vendittelli, M.: Real-time map building and navigation for autonomous robots in unknown environments, *IEEE Transaction Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 28, no. 3, pp. 316–333, 1998
- Sharma, D.K.: McDonald's problem—an example of using Dijkstra's programming method. *Bell Syst. Tech. J.* **60**(9), 2157–2165 (1981)
- Che, X., Zong, S., Che, N., Gao, Z.: The product calculation of linear polynomial and B-spline curve, *IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, 2009. *CAID & CD 2009*, pp. 991–994, 2009
- Abbas, T.F.: A bezier curve based free collision path planning of an articulated robot. *Eng. Tech. J.* **31**(7), 1263–1275 (2012)
- Ahmad A., Ali J.M.: Geometric control of rational cubic curve, *International Conference on Computer Graphics, Imaging and Visualization*, 2004. *CGIV 2004. Proceedings*, pp. 257–262, 2004
- Jiao, C.Y., Li, D.G.: Microarray image converted database—genetic algorithm application in bioinformatics, *BioMedical Engineering and Informatics*, 2008. *BMEI 2008. International Conference*, vol. 1, pp. 302–305, 2008
- Li, J., Qi Zha Y.: Genetic granular cognitive fuzzy neural networks and human brains for pattern recognition, *International Conference on Granular Computing*, 2005 *IEEE*, vol. 1, pp. 172–175, 2005
- Bin, H., Zhen, L.W., Feng, L.H.: The kinematics model of a two-wheeled self-balancing autonomous mobile robot and its simulation, *Second International Conference on Computer Engineering and Applications*, vol. 2, pp. 64–68, 2011
- Tu, A., Yi, M.: Dynamic path planning of mobile robots with improved genetic algorithm. *Sci. Dir. Comput. Electr. Eng.* **38**, 1564–1572 (2012)
- Ji, H. An hybrid heuristic algorithm for the two-echelon vehicle routing problem, *IET International Conference on Information Science and Control Engineering*, pp. 1–5, 2012
- Yu C.-J., Chen Y.-H., Wong C.-C.: Path planning method design for mobile robots, *SICE Annual Conference*, Tokyo, pp. 1681–1686, 2011

22. Neeraj, Ku A., Efficient hierarchical hybrid parallel genetic algorithm for shortest pathrountne, *Confluence the Next Generation Information Technology Summit International Conference*, 2014
23. Chalco-Cano, Y.: Fuzzy differential equations and Zadeh's extension principle, *Fuzzy Information Processing Society Annual Meeting of the North American*, pp. 1–5, 2011.
24. Ta, P.-S., Lin, Y.-H., Ta, C.-F.: Implementation of trajectory tracking control for wheeled mobile robots based on fuzzy controller. *J. China Univ. Sci. Technol.* **41**, 45–65 (2009)



Wei-Jen Chen was born in Kaohsiung, Taiwan, in 1987. He received the B.S. degree in electrical engineering from Fu Jen University, Shinjo, Taiwan, in 2013. He is currently working for his M.S. degree in Mechatronics Engineering, National Taiwan Normal University, Taipei, Taiwan. His areas of research interest include motion control of wheeled mobile robots, path planning, path tracking, and intelligent algorithms.



Bing-Gang Jhong was born in Taipei, Taiwan, in 1988. He received the B.S. and M.S. degrees in Mechatronics Engineering from National Taiwan Normal University, Taipei, Taiwan, in 2015. He is currently working for his Ph.D. in Mechatronics Engineering. His areas of research interest include robotics, adaptive control, fuzzy control, and machine vision.



Mei-Yung Chen received the B.S. degree from TamKang University in 1992, the M.S. degree from Chung Yuan Christian University in 1994, and the Ph.D. degree from National Taiwan University in 2003. He is currently a professor in the Department of Mechatronic Engineering, National Taiwan Normal University, Taiwan. His areas of research interest include engineering education, magnetic levitation, positioning and tracking, mechatronics, and advanced control theory and its applications.