J Electr Eng Technol.2017; 12(2): 918-925
http://doi.org/10.5370/JEET.2017.12.2.918

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Fuzzy-based Path Planning for Multiple Mobile Robots in Unknown Dynamic Environment

Ran Zhao* and Hong-Kyu Lee†

Abstract – This paper presents a path planning problem for multi-robot system in the environment with dynamic obstacles. In order to guide the robots move along a collision-free path efficiently and reach the goal position quickly, a navigation method based on fuzzy logic controllers has been developed by using proximity sensors. There are two kinds of fuzzy controllers developed in this work, one is used for obstacle avoidance and the other is used for orientation to the target. Both static and dynamic obstacles are included in the environment and the dynamic obstacles are defined with no type of restriction of direction and velocity. Here, the environment is unknown for all the robots and the robots should detect the surrounding information only by the sensors installed on their bodies. The simulation results show that the proposed method has a positive effectiveness for the path planning problem.

Keywords: Path planning, Fuzzy logic, Multi-robot, Moving obstacles, Robot navigation

1. Introduction

As an important branch subject of robotics, autonomic mobile robot technology has a long history and will be used widely in future. Mobile robots have vast application prospects in areas including space exploring, factory automation, mining, eliminating dangerous situation, military and service, which can economize the labor force to be engaged in other aspects. In these applications, the navigation problem of mobile robots is one of the most popular issues. Thus, how to detect the surrounding information and finding a safe path for the robot is the first condition of success. In an environment with obstacles, the navigation of a robot together with its path planning is to find the collision-free path from a starting location to a target location. The robots have several difficult problems which need to be solved, such as obstacle avoidance, position identification, driving safety, etc. Therefore, an intelligence algorithm can be used to provide a mobile robot with the ability to identify the current location, avoid any collisions and determine a suitable path to the goal position.

In the past few decades, plenty of researchers have contributed themselves to solve the path planning problem and a variety of methods have been proposed, such as neural network [1, 2], genetic algorithm [3], artificial vision method [4] and PID control [5] in static environment. These methods are usually used in global path planning and hardly be used in real-time control. Many other real-time control methods applied to static environment have also been developed. Researchers in [6] proposed a navigation method for mobile robots based on an improved remote control algorithm which combines the inertial sensors with vision information. In [7], researchers developed an expanded guide circle method for the navigation of mobile robots and improved the efficiency of the remote operation. Many traditional methods designed for global (static) path planning have also been extended to local (dynamic) path planning, such as potential field method [8], roadmap method [9] and rolling window method [10], etc. These methods can be performed well in static environment, but part of them usually are somewhat flawed in dynamic environment because of the assumptions for the moving obstacles as static in a certain period of time.

Navigation problem for multi-robot systems is also another issue for researchers. Tiago [11] has studied a formation control problem for multiple robots by using the Takagi-Sugeno type fuzzy automaton. Zhong [12] established a new Velocity-Change-Space method by using the changes of the speed and direction of the robot's velocity for dynamic motion planning. But all the robots must be measured the size, positions and velocities of obstacles and other robots online. Dayal [13] also studied the navigation problem for multiple robots by using Neuro-Fuzzy method. Similarly, an online inter-communication among robots is required. Therefore, come studying the proposed problem in completely unknown environment more seem to be have the necessity.

In this paper, we focus on a fuzzy-based path planning approach for multi-robot system in completely unknown dynamic environment. By means of the danger coefficient detected by a set of proximity sensors and the fuzzy inference system, the robots can avoid both static and dynamic obstacles and then seek for the target by another

† Corresponding Author: Dept. of Electrical Engineering, Korea University of Technology and Education, Korea. (hongkyu@koreatech.ac.kr)

* Dept. of Electrical Engineering, Korea University of Technology and Education, Korea. (zhaoran0922@163.com)

Received: April 14, 2015; Accepted: November 22, 2016

fuzzy controller step by step. In addition, all the robots have no difference and use the same navigation strategy.

The rest of this paper is organized as follows: Section 2 presents the modeling of a mobile robot simulated in this work and the kinematic functions. Section 3 describes the design of two kinds of fuzzy logic controllers. In section 4, we discuss the special situations in the procedure of obstacle avoidance. In Section 5, we verify the effectiveness of the proposed algorithm by a serious of simulations. Finally, conclusions and discussions are included in Section 6.

2. Modeling of Wheel Mobile Robot

In this paper, we choose a classic wheeled mobile robot as an example for simulation. There are two DC motors installed on left and right wheels respectively and both sides have an encoder on each one, so that the robot can detect how far each wheel has moved by itself. Usually there is also a gyro sensor installed on the robot body for feedback information and correction of driving direction in practice. The structure of such a robot with circular shape is as shown in Fig. 1. Totally there are 16 proximity sensors (ultrasonic sensors or infra-red sensors) equably installed around the robot with the same interval. These sensors which are numbered from s_0 to s_{15} are used to detect the distance to the surrounding obstacles or the other robots. We assume that the positions of all the obstacles and the other robots are unknown for the robot.

The data that defines a path element can be denoted as a series of coordinate values defining points. Let P_W and P_R be the center point between two wheels and the center of robot. We use the generalized coordinates $[\mathrm{x}, \mathrm{y}, \theta]^T$ to describe the configuration of every point. Thus, the kinematic functions of P_W can be given as:

$$\begin{cases} \dot{x} = \dfrac{v_L + v_R}{2}\cos\theta \\ \dot{y} = \dfrac{v_L + v_R}{2}\sin\theta \\ \dot{\theta} = \dfrac{v_L - v_R}{L} \end{cases}, \tag{1}$$

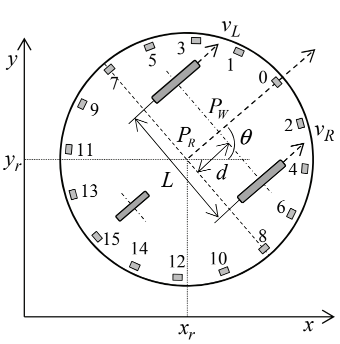


Fig. 1. Modeling of a wheel mobile robot

where v_L and v_R are the corresponding linear velocity of left and right wheels. θ denotes the angle between the robot's driving direction and x-axis and L is measured as the distance between two wheels.

We assume that the robot is driven under the condition with no slippage, thus, the robot should follow the nonholonomic constraint:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0\,. \tag{2}$$

Measured d as the length between P_W and P_R. According to the geometrical method, the kinematic function of P_R can be easily transformed from (1) as:

$$\begin{cases} \dot{x} = \dfrac{v_L + v_R}{2}\cos\theta - d\cos\theta \\ \dot{y} = \dfrac{v_L + v_R}{2}\sin\theta - d\sin\theta \\ \dot{\theta} = \dfrac{v_L - v_R}{L} \end{cases}. \tag{3}$$

3. Design of Fuzzy Logic Controller

The fuzzy logic control theory is one of the most successful areas in the application of intelligent control. It has shown excellent performance when the procedures are too complex for analysis by traditional mathematical description. The process where the input variables are treated in using a system, called a fuzzy logic controller (FLC). A FLC contains three processes which are fuzzification, fuzzy inference and defuzzification.

Our purpose is to guide the mobile robots with a collision-free path to move in a dynamic environment with moving objects and reach the target position successfully. Then, the path planning problem can be considered as obstacle avoidance and orientation to target. Thus, two kinds of FLCs with different effectiveness are developed in this paper.

3.1 Obstacle avoidance

Firstly, at the start position, the robot will move ahead with the direction towards the target. Once the sensors installed on the robot body detect any object (obstacles or other robots), an avoidance strategy is to be activated. This strategy is obtained by means of a fuzzy logic controller which is named as OA-FLC, based on a set of rules.

In this paper, we use the velocity vector $\vec{v}$ to denote the velocity of robots and moving obstacles, that is, $\vec{v} = [v,\ \theta]^T$. As shown in Fig. 2, P_R and P_O denote the current position of a robot and an obstacle and with the current velocity $\vec{v}_r$ and $\vec{v}_o$ respectively. The relative velocity of the robot with respect to the obstacle is

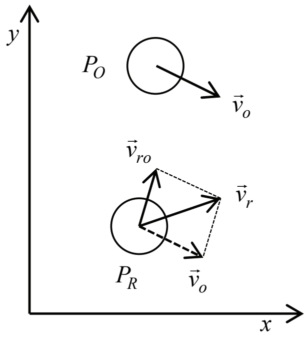


Fig. 2. Construction of relative velocity

$\vec{v}_{ro} = \vec{v}_r - \vec{v}_o$. In this figure, the direction of vector $\vec{v}_{ro}$ is towards the obstacle. It means that, if nothing is done now, there will be a collision between the robot and the obstacle.

The sensors installed on the robot body are set to run once every 0.1 second. Thus, in such a short time, we can simplify the relative velocity calculation as the following equation to obtain the acceptable approximate solutions:

$$v_{ro}^{i} \approx \frac{d_{t1}^{i} - d_{t2}^{i}}{\Delta t}, \tag{4}$$

where d_{t1}^{i} and d_{t2}^{i} are the distance detected by the i_{th} sensors ($i = 0 \cdots 15$) at time $t1$ and $t2$. And the time interval Δt of $t1$ and $t2$ is 0.1 second. We use the expected time of collision to denote the danger coefficient, that is:

$$R_{dc}^{i} = {d_{t2}^{i}} \big/ {v_{ro}^{i}}. \tag{5}$$

The smaller the value of R_{dc}^{i} is, the more dangerous the robot will be. Specially, when the direction of vector $\vec{v}_{ro}^{i}$ points away from the obstacle the value of R_{dc}^{i} is set as infinity.

Here, the minimum danger coefficient and the angle of the corresponding sensor are taken as the input variables of OA-FLC. Moreover, the linear velocity of left and right wheels will be the output variables. We can denote that:

$$\mathbb{R}_{dc} = \left\{ R_{dc}^{i} \mid 0 \le i \le 15 \right\}. \tag{6}$$

Then the input variables of OA-FLC can be described as:

$$R = \min(\mathbb{R}_{dc}), \tag{7}$$

$$\Phi = \angle s_i, \ when \ R_{dc}^{i} = \min(\mathbb{R}_{dc}), \tag{8}$$

where $-180^\circ \le \angle s_i < 180^\circ$, which is the corresponding angle of the i_{th} sensor.

The fuzzification process of a FLC transforms a non-fuzzy input values to a fuzzy value. For fuzzifier, the input and output variables can be divided into several linguistic terms as follow:

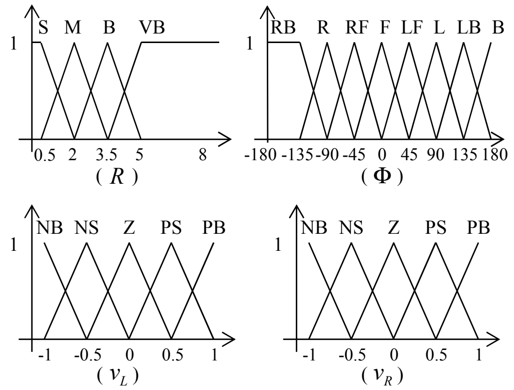


Fig. 3. Membership functions for input and output variables of OA-FLC

S: *small*; M: *medium*; B: *big*; VB: *very big*;
RB: *right back*; R: *right*; RF: *right front*; F: *front*;
LF: *left front*; L: *left*; LB: *left back*; B: *back*;
NB: *negative big*; NS: *negative small*; Z: *zero*;
PB: *positive big*: PS: *positive small.*

The transformation process of no-fuzzy input values to fuzzy values is achieved by using the membership functions which provide fuzzy terms with a definite meaning. Every fuzzy set which is used to specify a certain variable in terms of fuzzy values is defined by means of a membership function. The degree of membership of a no-fuzzy input value to a certain fuzzy set represents the confidence, expressed as a number from 0 to 1, that a particular value belongs to this fuzzy set [14].

The membership functions for input and output variables of OA-FLC are as shown in Fig. 3. As shown in this figure, when the main risk comes from left side the angle Φ is positive, otherwise it is negative. Moreover, we use the triangular type of membership functions with the range of $[-1, 1]$ for the output variables, which denote the scale of the maximum linear velocity.

The procedure of fuzzy inference usually contains a set of rules which are used to appoint the desired control behavior. A rules set is a condition description taking the form of "IF…THEN…" rules. The rules set applied for OA-FLC are as shown in Table 1. In this paper, the optimization problems (minimal path and time) will not be considered. For example, when the direction of impending danger is 'front', after the OA-FLC is activated, instead of turning toward the target to shorten the path length, the robot is fixed to turn left.

3.2 Target orientation

When there is no risk of collision between the robot and the other obstacles include both static and dynamic objects, or the robot has been driven over the obstacles temporarily,

Table 1. Rules set for OA-FLC

v_L		Φ							
		RB	**R**	**RF**	**F**	**LF**	**L**	**LB**	**B**
R	**S**	PB	PB	NB	NB	NS	PB	PB	PS
	M	PB	PB	NB	NB	PB	PB	PS	PS
	B	PB	PB	NS	Z	PB	PB	PS	PS
	VB	PB	PB	PS	PS	PB	PB	PS	PS

v_R		Φ							
		RB	**R**	**RF**	**F**	**LF**	**L**	**LB**	**B**
R	**S**	PB	PB	NS	NS	NB	PB	PB	PB
	M	PS	PB	PB	Z	NB	PB	PB	PB
	B	PS	PB	PB	PB	NS	PB	PB	PB
	VB	PS	PB	PB	PB	PS	PB	PB	PB

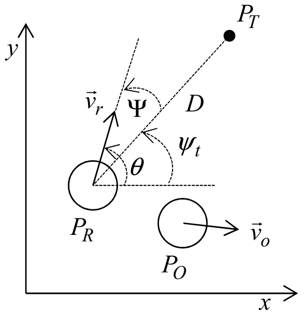


Fig. 4. Scheme of target orientation

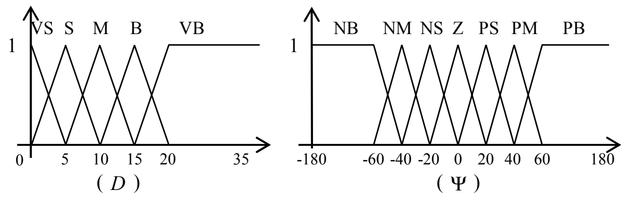


Fig. 5. Membership functions for input variables of TO-FLC

another strategy for target orientation will be activated. The strategy is obtained by another fuzzy controller, which is named TO-FLC.

The schematic model of target orientation is given in Fig. 4. In this figure, with the current position P_R , the robot has moved over the obstacle P_O . With the angle ψ_t between the line from the center of the robot to the target point and the x-axis, we denote that the angular difference is $\Psi = \theta - \psi_t$. Now, the obstacle is moving far away from the robot and poses no risk to the robot. Thus, the strategy of TO-FLC is come into use. This controller is used to drive the robot to point the target position. In the other words, it is used to reduce the angular difference Ψ which is one of the input variables of TO-FLC.

Another constraint for TO-FLC is the distance *D* between the robot and the target, which is denoted with the multiple of robot's radius in this paper. Moreover, the output variables are also the linear velocity of left and right wheels, which are the same with OA-FLC.

The membership functions for input variables of TO-FLC and the corresponding rules set are as shown in Fig. 5 and Table 2.

Table 2. Rules set for TO-FLC

v_L		Ψ						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
D	**VS**	NB	NB	NS	PS	PS	PB	PB
	S	NB	NS	Z	PB	PB	PB	PB
	M	NB	NS	Z	PB	PB	PB	PB
	B	NB	NB	NS	PB	PB	PB	PB
	VB	NB	NB	NB	PB	PB	PB	PB

v_R		Ψ						
		NB	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
D	**VS**	PB	PB	PS	PS	NS	NB	NB
	S	PB	PB	PB	PB	Z	NS	NB
	M	PB	PB	PB	PB	Z	NS	NB
	B	PB	PB	PB	PB	NS	NB	NB
	VB	PB	PB	PB	PB	NB	NB	NB

4. Special Situations in Obstacle Avoidance

In the environment with multiple dynamic obstacles, when the value of expected time of collision is smaller than "medium", if there are several obstacles in front of the robot and the main risk comes from straight ahead at the same time, or if there are several obstacles move toward the side of the robot, in such complex situations only use "IF…THEN…" rules is obviously not enough. This section discussed the special situations when the main risk comes from straight ahead, straight astern, directly left or directly right. When the value of expected time of collision is larger than "medium", the method developed in this section will not be used.

We use $\phi = \angle(\vec{v}_{ro} - \vec{v}_r)$ to denote the direction of coming risk. $\angle s_i$ denotes the angle of the i_{th} sensor. Then the special situations can be defined as follows:

straight ahead: if $\phi \subset \{\angle s_0, \angle s_1, \angle s_2\}$;
straight astern: if $\phi \subset \{\angle s_{13}, \angle s_{14}, \angle s_{15}\}$;
directly left: if $\phi \subset \{\angle s_5, \angle s_7, \angle s_9\}$;
directly right: if $\phi \subset \{\angle s_6, \angle s_8, \angle s_{10}\}$.

Two of four special situations are described as shown in Fig. 6. In dealing with the situation a), we will consider the difference of the distance between the robot and the obstacle which located at the left-front side of robot and such distance to the right-front side obstacles. Define that the distance difference of left-front and right-front side is:

$$D_{dif}^1 = \frac{s_1 + s_3}{2} - \frac{s_2 + s_4}{2} . \tag{9}$$

Then the constraint can be described as: if $D_{dif}^1 > 0$ the robot turns left, otherwise turns right.

In the case of b) in Fig. 6, the main risk comes from

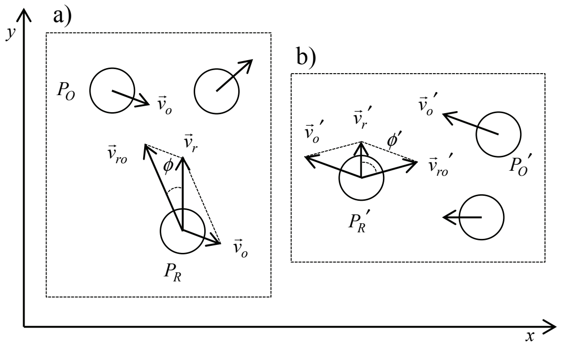


Fig. 6. Special situations when the main risk comes from straight ahead (a) and directly right (b)

Table 3. Navigation process for path planning

```
Algorithm: Navigation algorithm for path planning
Input: $R$, $\Phi$; $D$, $\Psi$
Output: $v_L$, $v_R$

Initialize: data of all sensors $s_i$; $D_{safe}$, $T_{safe}$ ← safety threshold
            of distance and expected time of collision
while the robot is not reached the target position do
    $R \leftarrow \min(\mathbb{R}_{dc})$
    while $R_{dc}^i = \min(\mathbb{R}_{dc})$ find
        $\Phi = \angle s_i$
    if $\min(s_i) < D_{safe}$ then
        if $R < T_{safe}$ then
            if $R <$ "medium" then
                if special situation mode then
                    $\Phi$ ← corresponding value
                end if
                do OA-FLC
            end if
            do OA-FLC
        else
            $D$, $\Psi$ ← generated by the data of encoders and (3)
            do TO-FLC
        end if
    else
        $D$, $\Psi$ ← generated by the data of encoders and (3)
        do TO-FLC
    end if
    update the coordinate of the robot by the data of encoders
```

right side. Moving forward or backward has become an important problem. Similarly, we denote that such distance difference of right-front and right-back side is:

$$D_{dif}^2 = \frac{s_4 + s_6}{2} - \frac{s_{10} + s_{12}}{2}. \tag{10}$$

Likewise, the constraint can be described as: if $D_{dif}^2 > 0$ the robot moves forward, otherwise moves backward. In the same way, the method discussed above can be easily extended to the situation of straight astern and directly left.

According to the method discussed above and the proposed FLCs designed in Section 3, the navigation process for path planning can be described as Table 3.

5. Simulation Results

In order to verify the effectiveness of the proposed algorithm, a series of simulations have been implemented with a set of Matlab codes for computing. We use R_i and T_i to denote the i_{th} robot and its corresponding target position. O_i and M_i are used to denote the i_{th} static and dynamic obstacles respectively. All the robots will move on a horizontal plane with the size of 100×100 unit-length. All the robots have the same size with the radius of 4 unit length. Every robot uses the same navigation strategy and the maximum velocity of left and right wheels is 4 unit length per second. The obstacles are approximated as circular sharp and the environments with no obstacles, static obstacles and both static and dynamic obstacles are simulated in this section. In addition, we will record the position of the robots by every one second to generate the path graph.

5.1 Results in no obstacle environment

The result of path graph in the environment with no obstacles is as shown in Fig. 7. There are three robots driven from the start points [0,0], [100,100], [0,50] to the target points [100,100], [0,0] and [100,50] respectively. All the robots moved ahead with the maximum speed until arrived the dangerous area (black dotted box), then the strategy of OA-FLC was activated.

Fig. 8 shows the velocity variation curve of left and right wheels for the robots. The areas in the black dotted boxes describe the velocity changes during the process of obstacle avoidance. That is, once the robot detect any other objects are approaching, it will slow down the speed immediately and adjust the direction to left or right by the

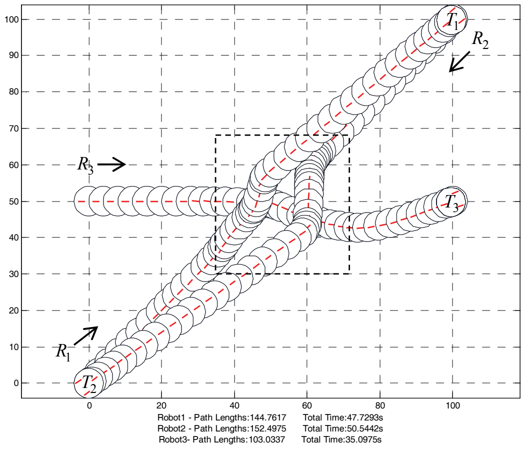


Fig. 7. Result in no obstacle environment

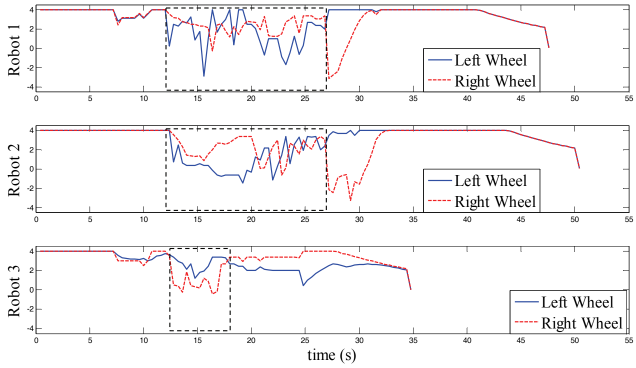


Fig. 8. Velocity variation curves of left and right wheels in no obstacle environment

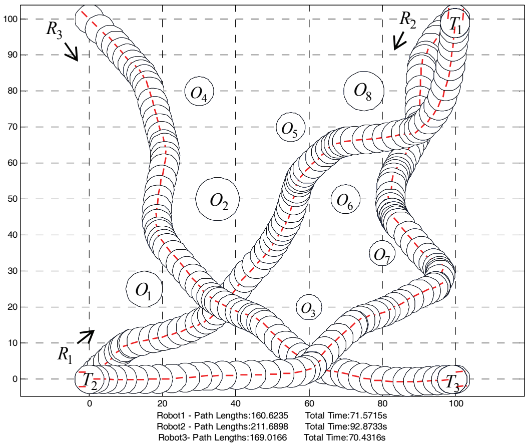


Fig. 9. Result in static obstacles environment

strategy of OA-FLC, then after leaving the dangerous area the velocity of left and right wheels are adjusted to towards the target point and raised to the maximum velocity again.

5.2 Results in static obstacles environment

The start and target points for all the robots and the final path graph are as shown in Fig. 9. All the obstacles in this paper are set to be circular sharp and have different size of radius. From this figure we can see that the proposed algorithm has good performance in such environment with multiple static obstacles.

The velocity change curves of left and right wheels for all the robots are shown in Fig. 10. The areas when the robots trying to avoid the obstacles which numbered from O_1 to O_8 have also been marked in this figure. In these marked areas, the robots find the correct directions through adjusting the velocity difference of left and right wheels to turn left or right according to the surrounding environment by OA-FLC strategy. And in the other unmarked areas the target orientation strategy is come into use.

5.3 Results in dynamic obstacles environment

The simulation results in the dynamic environment are as shown in Fig. 11. Here, there are two static and four dynamic obstacles. The dynamic obstacles are given different radius, driving directions and variable velocities, and will keep on moving between the certain points during the whole navigation process. As shown in Fig. 11(a), the trajectories of all the dynamic obstacles are denoted as dashed lines. In this simulation, dynamic obstacles M_1, M_2 move along the straight lined trajectories, and M_3, M_4 move along the triangular trajectories which formed by three vertices, respectively, at three different velocities between three vertices. The coordinate of the range of movement for dynamic obstacles and the corresponding velocity are defined as in Table 4. Here, the velocity is defined as the multiple of maximum linear velocity of the robot.

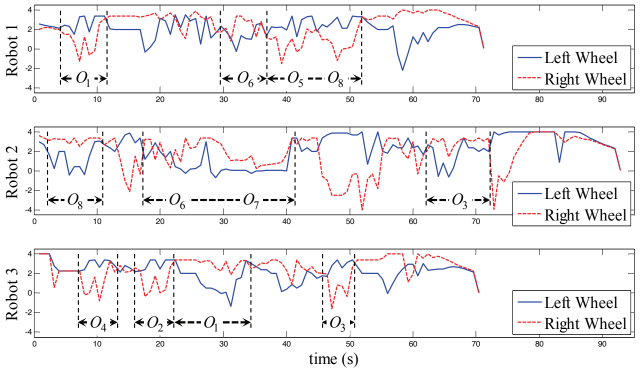


Fig. 10. Velocity variation curves of left and right wheels in environment with static obstacles

Table 4. Definition of the dynamic obstacles

Obstacles	Radius	Vertices of trajectories	Speed
M_1	3.5	[20,70] - [50,20]	0.9
M_2	4	[70,90] - [70,30]	0.3
M_3	3	[25,40] - [90,70] - [70,15]	0.75→0.3→0.35
M_4	4	[45,25] - [100,35] - [40,80]	0.7→0.6→0.3

As shown in Fig. 11, the robots started moving at frame a) and ended at frame j). The other frames describe the moments when avoiding obstacles and the black arrow represents the driving direction of the robots and obstacles at the corresponding moments. Specially, if it's impossible for the robot to avoid collision just by turning left or right, it will slow down until a complete stop (e.g., R1 at frame i and R3 at frame f, etc.), in some cases, it can also be able to move backwards. In the complex situations, the robot will not only use the fuzzy logic to avoid obstacles but also need to apply the rules mentioned in Section 4. For example, as shown from Fig. 11 d) to h), three obstacles were barreling toward R2 and R3, and two robots were also fast approaching to each other. Firstly, at frame d, both robots were slowed down rapidly to reduce the risk of collision. Then, at frame e, R3 continued moving ahead at low speed and R2 was guided to speed by. After R2 passed through the danger area from frame f, R3 was speeded up to pass through this area as quickly as possible. Moreover,

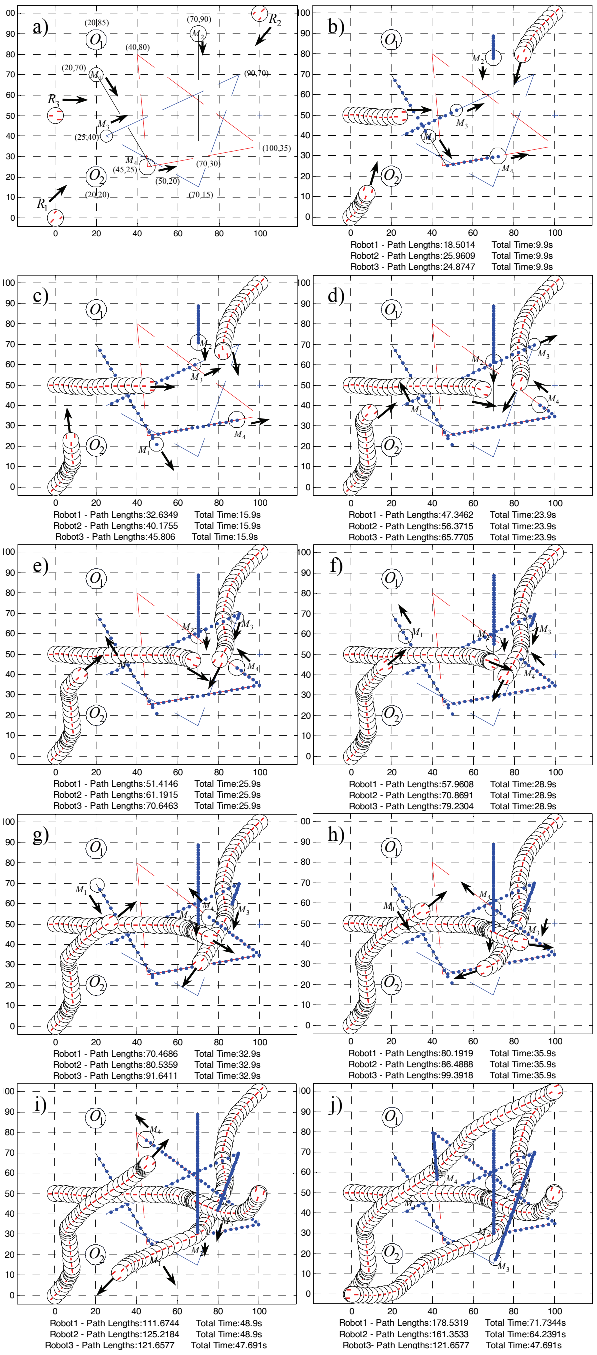


Fig. 11. Results in dynamic obstacles environment

the velocity change curves for all the robots are described as in Fig. 12 and the corresponding areas of obstacle avoidance have also been marked in this figure.

6. Conclusion

This paper studied a path planning problem for multiple mobile robots in the unknown environment with dynamic obstacles based on fuzzy logic controller. In this paper, both static and dynamic obstacles with different size and speed were investigated. The simulation provided satisfactory results for the proposed problem and the fuzzy approach has been verified to be effective.

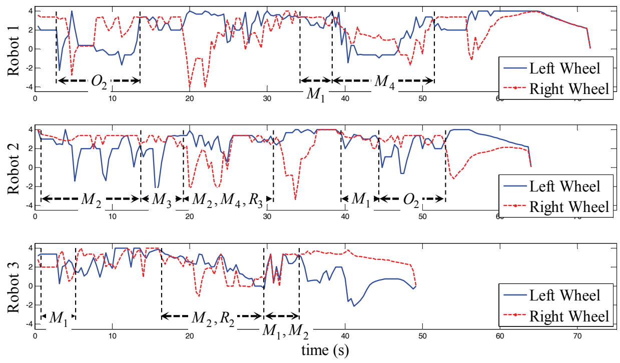


Fig. 12. Velocity variation curves of left and right wheels in environment with dynamic obstacles

The performance of a FLC is influenced by its rules set and the membership functions, which is indispensable to be adjusted by other optimization process, such as a genetic algorithm or a neural network, to get better performance and higher efficiency. The optimization problems of minimal path and time are not considered in this paper, to integrate the two FLCs and get the optimal path will be a meaningful work in the following research. In this paper, the dynamic obstacles move under preset trajectories, to perform the simulation in the environment with randomly moving obstacles will be a rewarding challenge for the future work. In addition, all the obstacles in this work are simulated as roundness, thus, in the future we will extend our experiment to the environment with dynamic irregular obstacles.

References

[1] D. Janglová, “Neural Networks in Mobile Robot Motion,” *International Journal of Advanced Robotic Systems,* Vol. 1, No. 1, pp. 15-22, 2004.

[2] K. H. Low, W. K. Leow, Ang, M. H. Jr., “Integrated Planning and Control of Mobile Robot with Self-organizing Neural Network,” *in Proceedings of ICRA'02. IEEE International Conference on Robotics and Automation*, vo1. 4, pp. 3870-3875, 2002.

[3] C. E. Thomas, M. A. C. Pacheco, M. M. and B. R. Vellasco, “Mobile Robot Path Planning Using Genetic Algorithms in Foundations and Tools for Neural Modeling,” *Springer Berlin/ Heidelberg*, vol. 1606/1999, pp. 671- 679, 1999.

[4] R. Carelli, C.M. Soria and B. Morales, “Vision-based Tracking Control for Mobile Robots,” *in Proceedings of 12th International Conference on Advanced Robotics*, pp. 148-152, July 2005.

[5] P. K. Padhy, T. Sasaki, S. Nakamura and H. Hashimoto, “Modeling and Position Control of Mobile Robot,” *in*

Proceedings of 11th IEEE international Workshop on Advanced Motion Control, pp. 100-105, Mar. 2010.

[6] D. Q. Khanh, Y. S. Suh, “Mobile Robot Destination Generation by Tracking a Remote Controller Using a Vision-aided Inertial Navigation Algorithm,” *Journal of Electrical Engineering & Technology*, Vol. 8 No. 3, pp. 613-620, May, 2013.

[7] S. H. Park, G. W. Kim, “Expanded Guide Circle-based Obstacle Avoidance for the Remotely Operated Mobile Robot,” *Journal of Electrical Engineering & Technology*, Vol. 9 No. 3, pp. 1034-1042, Sep, 2014.

[8] S. S. Ge, Y. J. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous Robots,* Vol. 13, No. 3, pp. 207-222, 2002.

[9] Van Den Berg J. P., Overmars, M. H., “Roadmap-based motion planning in dynamic environments,” *Robotics, IEEE Transactions on,* Vol. 21, No. 5, pp. 885-897, 2005.

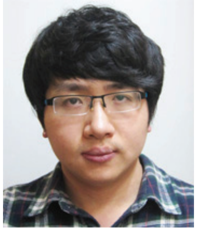
[10] C. G. Zhang, Y. G. Xi, “Rolling path planning and safety analysis of mobile robot in dynamic uncertain environment,” *Control Theory & Applications,* Vol. 20, No. 1, pp. 37-44, 2003.

[11] Tiago P. Nascimento, A. P. Moreira, A. G. S. Conceição, A. Bonarini, “Intelligent state changing applied to multi-robot systems,” *Robotics and Autonomous Systems,* Vol. 61, No. 2, pp. 115-124, Feb. 2013.

[12] X. Y. Zhong, X. G. Zhong, X. F. Peng, “Velocity-Change-Space-based dynamic motion planning for mobile robots navigation,” *Neurocomputing,* Vol. 143, No. 2, pp. 153-163, Nov. 2014.

[13] D. R. Parhi, S. K. Pradhan, A. K. Panda, R. K. Behera, “The stable and precise motion control for multiple mobile robots,” *Applied Soft Computing*, Vol. 9, No. 2, pp. 477-487, Mar. 2009.

[14] “Introduction to Fuzzy Systems,” in Proceedings of IEEE Electronic Technology Directions to Year 2000, pp. 94-103, May. 1995.



Ran Zhao He received his B.S. degree from the Department of Information Engineering at Shandong University, China, in 2006, and his M.S. degree from the Department of Electrical and Electronics Engineering at the Korea University of Technology and Education, Korea, in 2009. He is currently working toward a Ph.D. degree. His research interests include intelligent mobile robots, fuzzy logic control, and genetic algorithm.

Hong-Kyu Lee He received his B.S., M.S, and Ph.D. degrees from the Department of Electronics Engineering at Seoul National University, Seoul, Korea in 1977, 1979, and 1989, respectively. He was with the Agency of Defense Development from 1979 to 1992. Since 1992, he has been with the Department of Electrical Engineering, Korea University of Technology and Education, where he is currently a Professor. His major research interests include control theory and intelligent mobile robots, fuzzy logic control, neural networks, and genetic algorithms.