International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:5, No:6, 2011

Adaptive Path Planning for Mobile Robot Obstacle Avoidance

Rong-Jong Wai and Chia-Ming Liu

***Abstract*—**Generally speaking, the mobile robot is capable of sensing its surrounding environment, interpreting the sensed information to obtain the knowledge of its location and the environment, planning a real-time trajectory to reach the object. In this process, the issue of obstacle avoidance is a fundamental topic to be challenged. Thus, an adaptive path-planning control scheme is designed without detailed environmental information, large memory size and heavy computation burden in this study for the obstacle avoidance of a mobile robot. In this scheme, the robot can gradually approach its object according to the motion tracking mode, obstacle avoidance mode, self-rotation mode, and robot state selection. The effectiveness of the proposed adaptive path-planning control scheme is verified by numerical simulations of a differential-driving mobile robot under the possible occurrence of obstacle shapes.

***Keywords*—**Adaptive Path Planning, Mobile Robot Obstacle Avoidance

I. Introduction

IN In recent years, designs of autonomous car-like robots have received increased attention due to their potential applicability and usefulness in the automotive and robotics industry. The autonomy of robots depends on the capability of the robots to explore unknown environments. Different areas of research, such as parking [1], [2], navigation [3], trajectory tracking [4], and wall/lane-following [5], are all actively being investigated. Nowadays, mobile robots are a promising technology that will improve the quality of life by providing collision avoidance, reducing traffic gridlock, and allowing the replacement of dangerous tasks currently performed by human drivers. The path planning for the objective of obstacle avoidance [6], [7], which is one of the key issues in the mobile robot navigation, is the major topic to be investigated in this study.In general, the path planning can be divided into two categories: global path planning and local path planning. In the global path planning, the prior knowledge of the robot workspace should be available. Li *et al.* [8] focused on the design of the optimum path planning via the shortest distance criterion. Moreover, several optimization methods [9]–[11] have been developed to solve the optimum path planning problem. On the other hand, local path planning methods use ultrasonic sensors [12], [13], laser range finders [14], and on-board vision systems [15] to perceive the environment to perform on-line path scheduling. In this study, the workspace for the navigation of the mobile robot is assumed to be unknown and it has stationary obstacles only.In the local path planning, particular attention is paid to the local minima problem. This problem usually occurs when a robot navigates towards a desired target with no prior knowledge of the environment composed of concave obstacles or mazes and gets trapped in a loop. In order to get out of the loop, the robot must comprehend its repeated path passed through the same environment, e.g., it should memorize the environment. Wang and Liu [16] resolved the local navigation problem by recording the information of obstacle and trajectory into a "memory grid" of the environment. The grid-based map represents the robot workspace by a two dimensional array of elements as cells. However, it will spent sizable memory to store the robot workspace. Velagic *et al.* [17] presented the application of soft computing methodologies such as fuzzy logic, genetic algorithm and the Dempster-Shafer theory of evidence to a mobile robot navigation system. This scheme has the capability of obstacle avoidance with minimum distance according to the Dempster-Shafer evidence theory via a fusion of sensor data and map building, as well as a fuzzy logic path planning based on a modified potential field method. Except for the requirement of sizable memory, it will result in heavy computation burden as the implementation of the Dempster-Shafer evidence theory. Krishna and Karla [18] investigated a real-time collision avoidance to deal with the local minimum problem by classifying the environment based on the spatio-temporal sensory data sequences. Although this method has a good performance, it highly depends on the landmark recognition so that the exact coordination localization is required. Besides, it is also difficult to choose a correct way to follow the wall boundary. Wang and Liu [19] proposed a minimum risk method based on the trial-return behavior phenomenon. Even though the effect of this approach is favorable, it will waste high power during the trial-return motion and spend more time from the start point to the object. In this study, the main objective is to design an adaptive path-planning control scheme for the obstacle avoidance of a mobile robot without detailed environmental information, large memory size and heavy computation burden.

The authors are with the Department of Electrical Engineering, Yuan Ze University, Chung Li, Taiwan 32003, R.O.C. (phone: 886-3-4638800 ext 7117; fax: 886-3-4639355; e-mail: rjwai@saturn.yzu.edu.tw).

II. System Descriptions

The kinematics of a nonholonomic mobile robot is depicted in Fig. 1 [17], [20], [21]. It consists of a vehicle with two driving wheels mounted on the same axis, and a caster for carrying the mechanical structure. These two driving wheels are independently driven by two actuators for the motion and orientation, and this caster is a passive self-adjusted supporting wheel. It is assumed that this mobile robot is made up of a rigid frame equipped with non-deformable wheels, and they are moving in a horizontal plane. The distance of two driving wheels is $2b$, and both wheels have the same radius denoted by r. Point P is located in the intersection of straight line passing through the mass center (point C) of the mobile robot and an axis passing through two driving wheels. The

robotic movement is indicated by the motion of point P. The position of the robot in the global coordinate frame $\{O, U, V\}$ is completely specified by $(u\ v\ \theta)$, where u and v are the coordinates of the point P in the global coordinate frame. In Fig. 1, the variable θ is the orientation of the local coordinate frame $\{P, X, Y\}$ attached on the robot platform measured from the horizontal axis. For the mobile robot system considered here, the pure rolling and non-slipping condition states that this mobile robot can only move in the direction normal to the axis of the driving wheels. This assumption imposes a nonholonomic constraint on the robotic motion of the following form:

$$\dot{v}\cos\theta - \dot{u}\sin\theta = 0 \tag{1}$$

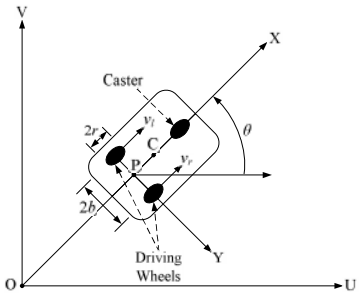


Fig. 1 Nonholonomic mobile robot

The dynamic model of the mobile robot in the global coordinate frame can be represented as [17]

$$\begin{aligned}\dot{u} &= (v_r + v_l)\cos\theta / 2 \\ \dot{v} &= (v_r + v_l)\sin\theta / 2 \\ \dot{\theta} &= (v_r - v_l)/2b\end{aligned} \tag{2}$$

where v_l and v_r are the linear velocities of the left wheel and the right wheel, respectively, and their values are bounded by a maximum velocity, $v_{\max}$. In this study, the control problem is to find a suitable path planning to produce a reference path trajectory for driving the mobile robot to reach its object safely.

III. Adaptive Path-Planning Control

In order to handle the safe movement of a nonholonomic mobile robot under different environment effectively, an adaptive path-planning control including motion tracking mode, obstacle avoidance mode, self-rotation mode, and robot state selection is proposed in this section, and the main flow chart is depicted in Fig. 2. Initially, the index Flag=1 is set, which means the robot with global coordinate (u, v) is tracking the object position (u_o, v_o). The robot moves directly toward the object according to the motion tracking mode as shown in Fig. 3 until the robot detects an obstacle. In this study, the robot perceives its environment through an array of eight sonar sensors. These eight sonar sensors on the robot can be grouped as left sensors (s_1, s_2, s_3 and s_4) and right sensors (s_5, s_6, s_7 and s_8). When the robot detects an obstacle (i.e., $\sum_{i=1}^{8} d_{mi} > 0$, where $d_{mi} = d_e - d_{si}$ represents the intensity of the ith sonar sensor, in which d_e is the maximum detectable distance by the sonar sensor, and d_{si} is the distance reading of the ith sonar sensor), it will operate at the obstacle avoidance mode (Flag=0) as shown in Fig. 4 to prevent colliding with the obstacle. Note that, the position control in Fig. 4 means to manipulate the robot via appropriate linear velocities of the left wheel and the right wheel to track the reference trajectory produced by the proposed adaptive path-planning control scheme. It could be a proportional-integral (PI) controller [17], a fuzzy logic controller [20] or a neural network controller [21]. In this study, a conventional PI control scheme is used.

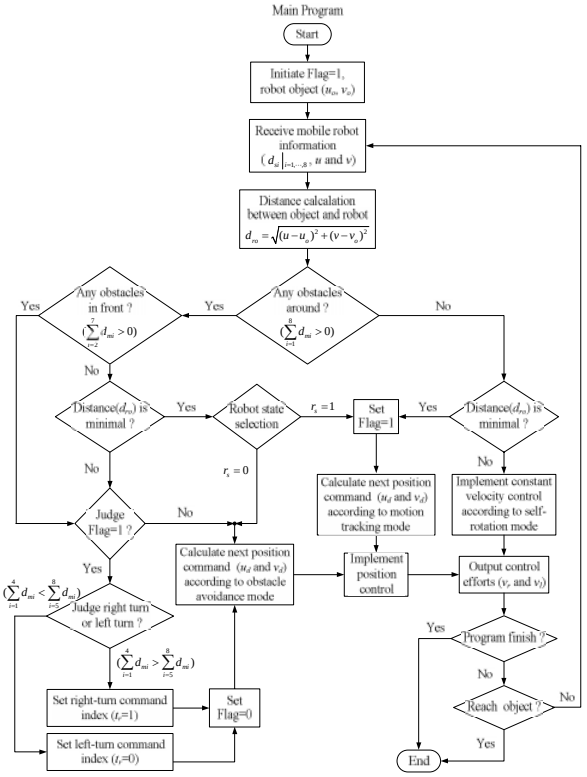


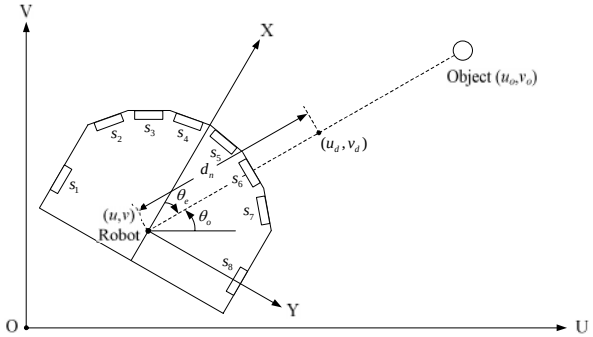
Fig. 2 Main flow chart of adaptive path-planning control

In order to distinguish the operational situations in the robot movement, the distance (d_{ro}) between the object and the robot is defined as

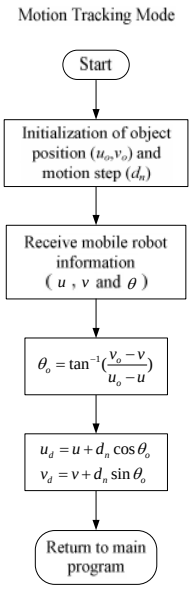
$$d_{ro} = \sqrt{(u-u_o)^2 + (v-v_o)^2} \tag{3}$$

There are two situations according to the variation of the distance (d_{ro}). In situation 1, the present distance is minimal after comparing the distances in the past, which means the robot is approaching to the object. On the other hand, after comparing the distances in the past, the present distance is not minimal (situation 2), which means the robot is leaving far away the object due to the operation of the obstacle avoidance mode. Because the sonar sensors only can detect 180 degree in the front, the obstacle may disappear at the rear of the robot during the obstacle avoidance mode in a short while. In order to resolve this problem, the operation of the self-rotation mode as shown in Fig. 5 is helpful to find the obstacle again in situation 2. In other words, the obstacle avoidance mode will

be operated when the obstacles are located around the robots. However, if it operates the obstacle avoidance mode continuously, the robot in situation 1 will miss the object. Therefore, the mechanism of the robot state selection as shown in Fig. 6 is designed to deal with this problem. That is, the robot needs to reselect to operate at the motion tracking mode or the obstacle avoidance mode when the robot lies in situation 1 and has obstacles in its surroundings. The detailed descriptions of the motion tracking mode, obstacle avoidance mode, self-rotation mode, and robot state selection are expressed in the following subsections.



(a)



(b)

Fig. 3 Motion tracking mode: (a) Sketch map; (b) Flow chart

A. Motion Tracking Mode

The diagram of the robot operated at the motion tracking mode and its flow chart are depicted in Fig. 3(a) and (b), respectively. The robot receives the information of the present robot posture (u,v,θ) from the wheel motion derived from encoder readings. The object angle (θ_o) measured from the horizontal axis can be represented as

$$\theta_o = \tan^{-1}(\frac{v_o - v}{u_o - u}) \tag{4}$$

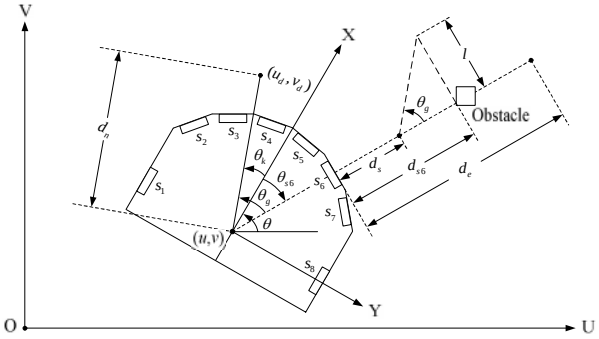
Thus, the desired robot position can be expressed as

$$\begin{aligned} u_d &= u + d_n \cos\theta_o \\ v_d &= v + d_n \sin\theta_o \end{aligned} \tag{5}$$

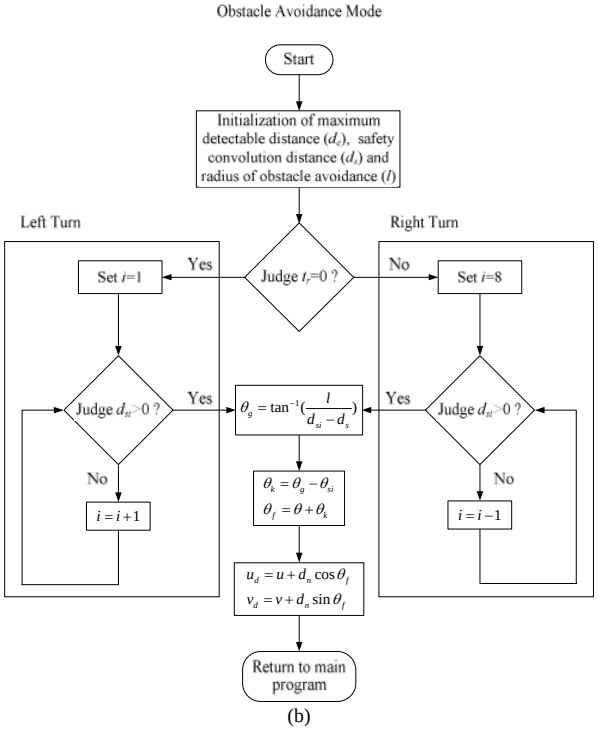
where d_n is the motion step designed by the user. In Fig. 3(a), the angle between the robot moving direction and the line connecting the robot center with the object can be given by

$$\theta_e = \theta - \theta_o \tag{6}$$

The task of the motion tracking mode is to calculate the next position command (u_d, v_d) with the absence of obstacles.



(a)



(b)

Fig. 4 Obstacle avoidance mode: (a) Sketch map; (b) Flow chart

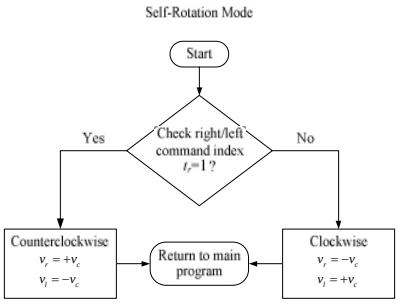


Fig. 5 Flow chart of self-rotation mode

B. Obstacle Avoidance Mode

The diagram of the robot operated at the obstacle avoidance mode and its flow chart are depicted in Fig. 4(a) and (b), respectively. When the obstacle around the robot is detected, the right/left command index (t_r) can be expressed as

$$\begin{aligned} &\text{if } (\sum_{i=1}^{4} d_{mi} < \sum_{i=5}^{8} d_{mi}), \text{ then } t_r = 0 \\ &\text{if } (\sum_{i=1}^{4} d_{mi} > \sum_{i=5}^{8} d_{mi}), \text{ then } t_r = 1 \end{aligned} \tag{7}$$

If the condition of $\sum_{i=1}^{4} d_{mi} < \sum_{i=5}^{8} d_{mi}$ holds, it means that the obstacle is close to the right side of the robot. Thus, one should set the right/left command index as $t_r = 0$ to make the robot turning left to avoid the obstacle. On the other hand, one should set the right/left command index as $t_r = 1$ to make the robot turning right to prevent collision if the condition of $\sum_{i=1}^{4} d_{mi} > \sum_{i=5}^{8} d_{mi}$ holds. In Fig. 4(b), the avoidable angle command can be represented as

$$\theta_g = \tan^{-1}(\frac{l}{d_{si} - d_s}) \tag{8}$$

where l is the radius of the obstacle avoidance to preset a region for eluding; d_s is the safety convolution distance of the robot. According to (8), the robot can prevent colliding with the obstacle and keep the safety convolution distance with the obstacle. Then, the angle to be rotated by the robot can be denoted as

$$\theta_f = \theta + \theta_g - \theta_{si} \tag{9}$$

where θ_{si} are the fixed angles of eight sonar sensors mounted on the robot. Thus, the desired robot command (u_d and v_d) can be given by

$$\begin{aligned} u_d &= u + d_n \cos\theta_f \\ v_d &= v + d_n \sin\theta_f \end{aligned} \tag{10}$$

Note that, the objective of the left-turn and right-turn areas in Fig. 4(b) is to search the distance reading (d_{si}) of the ith sonar sensor with the first detection of the obstacle by the direction of clockwise or counterclockwise for calculating the rotation angle (θ_f) of the robot.

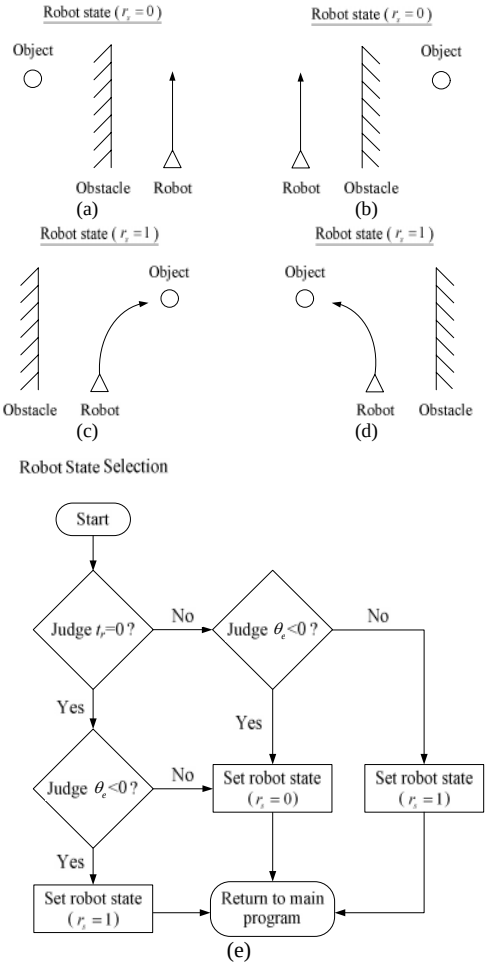


Fig. 6 Robot state selection: (a)–(b) Sketch map at robot state (r_s=1); (c)–(d) Sketch map at robot state (r_s=0); (e) Flow chart

C. Self-Rotation Mode

Because the obstacle may disappear at the rear of the robot during the obstacle avoidance mode in a short while, the operation of the self-rotation mode as shown in Fig. 5 is helpful to find the obstacle again. When the robot turns left to avoid the obstacle ($t_r = 0$), the robot needs the clockwise rotation to find the obstacle. On the other hand, the robot needs the counterclockwise rotation to get the obstacle when the robot turns right to avoid the obstacle ($t_r = 1$). In order to implement the self-rotation mode, the following constant velocity control is adopted in this study:

$$\text{Clockwise: } v_r = -v_c, \; v_l = +v_c \tag{11}$$

$$\text{Counterclockwise: } v_r = +v_c, \; v_l = -v_c \tag{12}$$

where v_c is a predetermined constant velocity. The operation of the self-rotation mode terminates until the robot finds the obstacle again, and then it will switch back to the obstacle avoidance mode.

D. Robot State Selection

If there are obstacles in the surrounding of the robot, but $\sum_{i=2}^{7} d_{mi} = 0$, one should re-judge to operate at the motion tracking mode or the obstacle avoidance mode because it may lose an opportunity to approach the object when it operates at the obstacle avoidance mode continuously. In order to distinguish the possible states, the locations of the obstacle, object and robot are illustrated in Fig. 6(a)–(d). When the location of the object and obstacle lies in the same side of the robot ($r_s = 0$), the robot should enter the obstacle avoidance mode. On the other hand, the robot should enter the motion tracking mode when the location of the object and obstacle lies in different sides of the robot ($r_s = 1$). The corresponding flow chart of the robot state selection is depicted in Fig. 6(e). The right/left command index $t_r = 0$ means that the obstacle is on the right side of the robot; otherwise, the obstacle is on the left side of the robot ($t_r = 1$). Moreover, if the angle $\theta_e < 0$ holds, the object is on the left side of the robot; otherwise, the object is on the right side of the robot. The robot should operate at the motion tracking mode as $r_s = 1$ because there is no obstacle between the object and the robot. On the other hand, the robot should operate at the obstacle avoidance mode as $r_s = 0$ because there has an obstacle between the object and the robot.

IV. Numerical Simulations

In this study, a differential-driving mobile robot is used for an example to verify the effectiveness of the proposed control scheme. The detailed parameters of this mobile robot are listed as follows:

$$r = 0.0925\,\text{m},\ b = 0.167\,\text{m},\ v_{\max} = 0.4\text{m/s},$$
$$\theta_{s1} = -90°,\ \theta_{s2} = -50°,\ \theta_{s3} = -30°,\ \theta_{s4} = -10°,\quad (13)$$
$$\theta_{s5} = 10°,\ \theta_{s6} = 30°,\ \theta_{s7} = 50°,\ \theta_{s8} = 90°$$

Moreover, the control parameters in the adaptive path-planning control are given as

$$d_n = 0.004\text{m},\ l = 0.1\text{m},\ \Delta t = 0.1\text{s},\ v_c = 0.04\text{m/s}\quad (14)$$

All the parameters in (14) are chosen to achieve the best transient control performance in numerical simulations by considering the possible operating conditions.All numerical simulations are carried out via the MATLAB software. In the simulations, four types of obstacle shape have been chosen to test the adaptability of the proposed control scheme. All the numerical simulations of the adaptive path-planning control scheme due to different obstacles are depicted in Figs. 7–10. Fig. 7 shows the responses at the object position ($u_o = 3, v_o = 5.5$) due to four square obstacles and three initial postures including $(u = 1.5, v = 1)$, $(u = 3, v = 1)$ and $(u = 4, v = 2.8)$. As can be seen from this figure, the robot can successfully avoid the obstacle and reach the object under different initial postures. Moreover, Fig. 8 illustrates the responses at the object position ($u_o = 3, v_o = 5.5$) due to U-shape obstacles and two initial postures including $(u = 3, v = 1)$ and $(u = 2.5, v = 2.5)$. In the first initial posture $(u = 3, v = 1)$, initially the robot moves directly toward the object due to the motion tracking mode until it reaches the point "a", where the robot detects an obstacle at the direct front. After that, it makes a right turn according to the obstacle avoidance mode until the robot reaches the point "b". When the robot locates at the point "b", the robot state is $r_s = 1$ and the distance (d_{ro}) is minimal after comparing the distances in the past. Therefore, the robot moves toward the object according to the motion tracking mode. In the second initial posture $(u = 2.5, v = 2.5)$, the robot moves toward the point "c" according to the obstacle avoidance mode. When the robot locates at the point "c", the robot enters the motion tracking mode to move toward the object because the robot state is $r_s = 1$ and the distance (d_{ro}) is minimal after comparing the distances in the past.

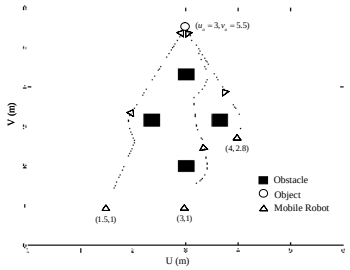


Fig. 7 Numerical simulations of adaptive path-planning control due to four square obstacles placed in the workspace

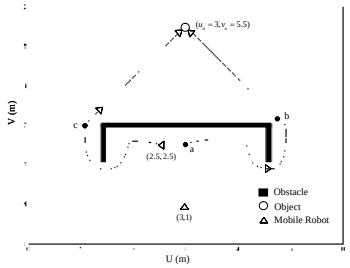


Fig. 8 Numerical simulations of adaptive path-planning control due to U-shape obstacles placed in the workspace

Fig. 9 illustrates the response at the object position ($u_o = 5, v_o = 5$) due to cluttered maze. By starting from the initial posture $(u = 0.5, v = 1.5)$, the robot moves by the obstacle avoidance mode to the point "a". In the point "a", the robot state is $r_s = 0$ and the distance (d_{ro}) is minimal after comparing the distances in the past, so that it makes a right turn according to the obstacle avoidance mode until the robot reaches the point "b". Because the distance between the point "b" and the object is smaller than the distance between the point "a" and the object, and the robot state is $r_s = 1$, the robot moves toward the point "c" according to the motion tracking mode. When the robot passes through the point "c", it switches to the obstacle avoidance mode until the robot reaches the point "d" because the robot state is $r_s = 0$ and the distance (d_{ro}) is minimal after comparing the distances in the past. Since there is no obstacle around the robot, and the distance (d_{ro}) from the point "d" to the point "e" is minimal, the robot moves directly toward the object according to the

motion tracking mode until the robot reaches the point "e", where the robot detects an obstacle at the direct front. When the robot passes through the point "e", the robot moves toward the point "f" according to the obstacle avoidance mode. When the robot reaches the point "f", the robot state is $r_s = 1$, and the distance between the point "f" and the object is minimal after comparing the distances in the past, so that the robot moves directly toward the object due to the motion tracking mode. Fig. 10 depicts the response at the object position ($u_o = 5, v_o = 5$) due to obstacles with different shape. By starting from the initial posture $(u = 0.5, v = 0.5)$, the robot can successfully pass through obstacles with different shape to reach the object.

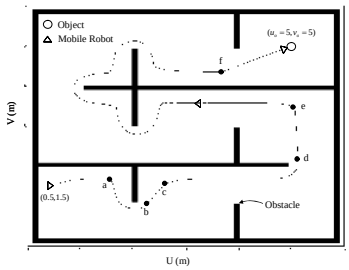


Fig. 9 Numerical simulations of adaptive path-planning control due to cluttered maze placed in the workspace

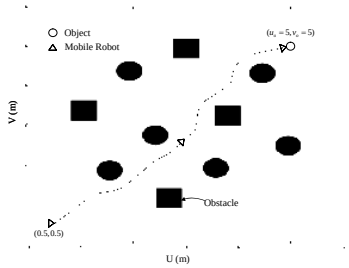


Fig. 10 Numerical simulations of adaptive path-planning control due to obstacles with different shape placed in the workspace

V. Conclusions

This study has successfully developed an adaptive path-planning control scheme to the obstacle avoidance of a mobile robot. Numerical simulations are given to verify the effectiveness of the adaptive path-planning control scheme. The major contributions of this study are recited as follows. 1) The successful incorporation of the motion tracking mode and obstacle avoidance mode into the adaptive path-planning control scheme to reach the object with obstacle avoidance. 2) The successful design of the self-rotation mode to prevent the obstacle disappearing at the rear of the robot during the obstacle avoidance mode in a short while. 3) The successful development of the robot state selection mechanism to hold an opportunity to approach the object during the obstacle avoidance mode. The proposed adaptive path-planning control scheme can be easily incorporated into other mobile robot navigation systems to provide an alternative for the objective of obstacle avoidance.

References

[1] T. C. Lee, C. Y. Tsai, and K. T. Song, "Fast parking control of mobile robots: a motion planning approach with experimental validation," IEEE Trans. Contr. Syst. Technol., vol. 12, no. 5, pp. 661676, 2004.

[2] T.-H. S. Li, S. J. Chang, and Y. X. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot," IEEE Trans. Ind. Electron., vol. 50, no. 5, pp. 867880, 2003.

[3] H. Seraji and A. Howard, "Behavior-based robot navigation on challenging terrain: a fuzzy logic approach," IEEE Trans. Robot. Automat., vol. 18, no. 3, pp. 308321, 2002.

[4] C. L. Hwang, L. J. Chang, and Y. S. Yu, "Network-based fuzzy decentralized sliding-mode control for car-like mobile robots," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 574585, 2007.

[5] W. Tsui, M. S. Masmoudi, F. Karray, I. Song, and M. Masmoudi, " Soft-computing-based embedded design of an intelligent wall/lane-following vehicle," IEEE/ASME Trans. Mechatronics, vol. 13, no. 1, pp. 125135, 2008.

[6] C. Ye, H. C. Yung, and D. Wang, "A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance," IEEE Trans. Syst. Man, Cybern. B, vol. 33, no. 1, pp. 17 27, 2003.

[7] J. H. Lilly, "Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle," IEEE Trans. Fuzzy Syst., vol. 15, no. 4, pp. 718728, 2007.

[8] Q. Li, W. Zhang, Y. Yin, Z. Wang, and G. Liu, "An improved genetic algorithm of optimum path planning for mobile robots," Int. Conf. Intelligent Systems Design and Applications, vol. 2, pp. 637642, 2006.

[9] J. Tu and S. Yang, "Genetic algorithm based path planning for a mobile robot," IEEE Int. Conf. Robotics and Automation, pp. 122111226, 2003.

[10] Y. Hu and S. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," IEEE Int. Conf. Robotics and Automation, pp. 435504355, 2004.

[11] W. Wu and Q. Ruan, "A gene-constrained genetic algorithm for solving shortest path problem," Int. Conf. Signal Processing, pp. 251025513, 2004.

[12] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," IEEE Trans. Robot. Automat., vol. 7, no. 3, pp. 278288, 1991.

[13] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," IEEE Trans. Syst. Man, Cybern. C, vol. 37, no. 4, pp. 610621, 2007.

[14] F. Amigoni and S. Gasparini, "Building segment-based maps without pose information," Proc. IEEE, vol. 94, no. 7, pp. 134013 59, 2006.

[15] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar genmetry," IEEE Trans. Robotics, vol. 23, no. 1, pp. 87100, 2007.

[16] M. Wang and J. N. K. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, " Robot. Autonomous Syst., vol. 56, no. 7, pp. 625643, 2008.

[17] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," Robot. Autonomous Syst., vol. 54, no. 12, pp. 98991004, 2006.

[18] K. M. Krishna and P. K. Kalra, "Perception and remembrance of the environment during real-time navigation of a mobile robot," Robot. Autonomous Syst., vol. 37, pp. 25–51, 2001.

[19] M. Wang and J. N. K. Liu, "Fuzzy logic based robot path planning in unknown environments," Int. Conf. Machine Learning and Cybernetics, vol. 2, pp. 813–818, 2005.

[20] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," IEEE Trans. Fuzzy Syst., vol. 15, no. 2, pp. 211–221, 2007.

[21] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach," IEEE Trans. Circuit Syst. I, vol. 53, no. 6, pp. 1381–1394, 2006.