

Real-Time Control Of Time-Critical Trajectory Planning For A Car-Like Robot In Unknown Environments

Mohd Sani Mohamad Hashim
*School of Mechatronic Engineering
Universiti Malaysia Perlis
Pauh Putra Campus
02600 Arau, Perlis
MALAYSIA*

Tien-Fu Lu
*School of Mechanical Engineering
University of Adelaide
North Terrace, Adelaide
South Australia 5005
AUSTRALIA*

Abstract

This paper presents a time-critical trajectory planning for a car-like robot. With the time-critical trajectory planning, the robot is expected to reach the final point at the specified travelling time. The trajectory is derived from a set of polynomial equations. Furthermore, the time taken during avoiding the obstacle will certainly delay the robot from reaching the final point at the specified travelling time. This problem is resolved by introducing a dynamic trajectory planning scheme for the obstacle avoidance approach, which will take into account the delayed time to replanning the trajectory after avoiding the obstacle. The algorithm was tested in the unknown static environment using a modified RC car as the car-like robot. The results demonstrate the performance of the proposed algorithm with the real-time control strategy for the robot.

1. Introduction

Research in trajectory planning for autonomous mobile robots has been popular over the past two decades. A car-like robot is one of the platforms that is widely investigated in this area [1,2,5,14-22]. Nonholonomic systems for car-like robots make trajectory planning more complicated. This type of robot, more than many other types of robots, is bounded by its kinematic and dynamic constraints which limit its motion. For example, such a robot can

only turn its steering wheels within certain limited degrees and is unable to make either a sharp turn or turn on spot. Moreover, its velocity and acceleration are also limited by its actuation and dynamics. A practical and effective trajectory planning approach would have to take all these limitations into account.

Furthermore, the question of timing in mobile robot navigation still remains an area of research that is not thoroughly investigated. Most research focuses more on optimising travel time or obtaining the minimum travel time [23,24] which may lead to the shortest path planning. However, in certain situations, a mobile robot may need not only to reach the desired location at optimum travel time, but to reach it at the specified time. Such situations may have significant ramifications for task-based applications such as patrolling large areas, exchanging and delivering goods, or coordinating a group of mobile robots. For all these applications, timing is important for the mobile robot. For example, in patrolling large areas, a mobile robot usually is assigned to complete patrolling the area within a specific time. Also it needs to pass through all the desired control points at a specific orientation in order to secure the area. Thus time-dependent trajectory planning is required to ensure the mobile robot is able to complete the patrol at the specified time. Similarly when two or more mobile robots are assigned to exchange and deliver goods, they may need to meet at a specific location and time in order to exchange the goods and then continue to deliver the goods to different locations. Hence the mobile robots do not wait for each other for a long time if the motion of each robot can be

controlled to reach the desired location at the specific time. Therefore, a trajectory planning approach for nonholonomic mobile robots is required to meet the criteria of time-critical and ability to pass through all the control points.

In this paper, the trajectory planning is developed based on a geometric approach [1-22]. The geometric approach has several advantages over other standard approaches such as roadmap, cell decomposition and potential fields method. The most significant advantage of the geometric approach is that the curvature can be represented as a function of time. This means the velocity and acceleration profiles can be easily derived from the generated path curvature. If the velocity and acceleration can be obtained, the motion of the mobile robot along the path can be controlled. Therefore the mobile robot will be able to reach the goal position with correct orientation at the specified time.

Furthermore, in every trajectory planning for the mobile robot, an ability to avoid an obstacle is crucial. Thus a dynamic trajectory planning scheme is also included to ensure mobile robots are able to avoid the obstacles smoothly and safely as well as to gain the time lost during obstacle avoidance. This obstacle avoidance approach is derived from [9]. However, the polynomial curves are used to perform this approach instead of Bezier curves. The reason behind this is to achieve a consistency in generating the trajectory, which derived from the original proposed trajectory planning.

2. Kinematic Model

The model of a nonholonomic mobile robot is shown in Figure 1. Consider a four wheel, rigid body robot, with front steering wheels and rear driving wheels, moving in a 2D environment. Both steering wheels are assumed to turn at the same degree and act as a single steering wheel.

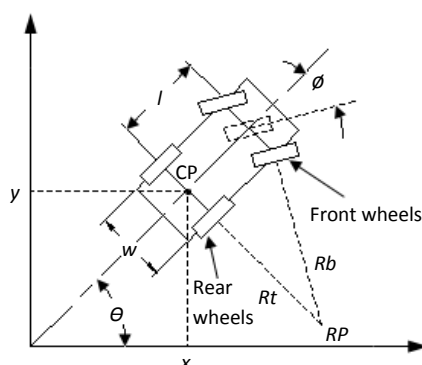


Figure 1. Mobile robot model

The robot state in Cartesian space can be represented by $q=[x, y, \theta, \phi, v, t]^T$, where (x, y) are the coordinate at the middle of the rear axle (CP), θ is the orientation of the robot with respect to the main axis, ϕ is the steering angle with respect to the robot orientation, v is the speed of the mobile robot and t is the time. The width of the mobile robot is given by w . Thus, the kinematic model of the mobile robot becomes

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \tan \phi / l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \rho u_1 & u_2 \end{bmatrix}^T \quad (1)$$

where,

l = wheelbase of the mobile robot

ρ = radius of rear wheel

u_1 = angular velocity of the driving wheel

u_2 = steering velocity of the steering wheel.

The following are some general assumptions made for this work:

- (i) The mobile robot moves on a planar surface,
- (ii) The wheels rotate without slipping,
- (iii) The wheels are not deformable,
- (iv) The steering rotates instantly.

Note that the turning radius of the mobile robot in this study is given by Rb , which is centred at RP . Due to geometric constraints of the mobile robot, the steering angle is limited to $-\phi_{max} \leq \phi \leq \phi_{max}$.

The trajectory generation is defined based on the geometric approach using polynomial curves. The general form of polynomials with degree n can be represented as:

$$f(x) = \sum_{j=0}^n c_j x^j \quad (2)$$

From $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \tan \phi / l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \rho u_1 & u_2 \end{bmatrix}^T$ (1), we have:

$$\frac{dy}{dx} = \tan \theta, \quad \frac{d^2 y}{dx^2} = \frac{\tan \phi}{l \cos^3 \theta} \quad (3)$$

Then the boundary conditions are set as:

$$\begin{aligned} q(t_0) &= q^0 = [x_0, y_0, \theta_0, \phi_0, v_0, t_0]^T, \\ q(t_f) &= q^f = [x_f, y_f, \theta_f, \phi_f, v_f, t_f]^T \end{aligned} \quad (4)$$

Thus by expanding (3) and (4), the trajectory can be generated by a set of equations as follows:

$$\begin{aligned} x &= a_0 + a_1t + a_2t^2 + a_3t^3, \\ y &= b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5 \end{aligned} \quad (5)$$

3. Obstacle Avoidance Approach

During avoiding an obstacle, the mobile robot will take some times to plan and avoid the obstacle. This situation will delay the travelling time for the mobile robot to reach the final point at the specified time. In order to encounter this problem, the obstacle avoidance approach is proposed based on the dynamic trajectory planning scheme, which has been adapted from the works in [9]. However, the polynomial curves are used instead of Bezier curves to achieve a consistency in trajectory planning for this work. The strategy to avoid a static obstacle for the mobile robot is illustrated in Figure 2.

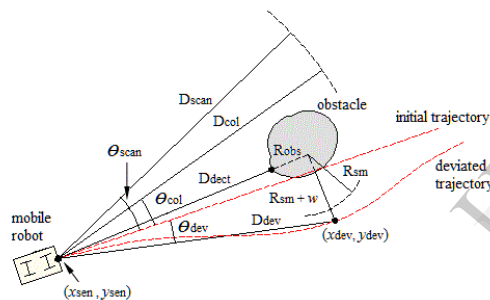


Figure 2. Avoiding a static obstacle

When the mobile robot starts to navigate along the initial trajectory, the range finder will also start to scan the environment. The maximum scanning range and resolution is set by D_{scan} and θ_{scan} , respectively. Once the mobile robot detects an obstacle, it will check whether the obstacle is within collision region or not. The collision region is defined by collision range (D_{col}) and collision angle (θ_{col}). If the obstacle falls into this region, a new deviated point will be determined in order to readjust the initial trajectory and to ensure the mobile robot avoids the obstacle. The deviated point (x_{dev}, y_{dev}) is determined by detection distance (D_{dect}), detection angle (θ_{dect}), obstacle's size (R_{obs}), safety margin (R_{sm}), robot's width (w) and sensor's position (x_{sen}, y_{sen}) . The following equations are used to obtain the deviated point:

$$d_{dev} = \sqrt{(R_{sm} + w)^2 + (D_{dect} + R_{obs})^2} \quad (6)$$

$$\theta_{dev} = \theta_{dect} + \tan^{-1} \left(\frac{R_{sm} + w}{D_{dect} + R_{obs}} \right) \quad (7)$$

$$x_{dev} = x_{sen} + d_{dev} * \cos \theta_{dev} \quad (8)$$

$$y_{dev} = y_{sen} + d_{dev} * \sin \theta_{dev} \quad (9)$$

Once the deviated point is obtained, a new trajectory (deviated trajectory) is generated from the current point to the final point, through the deviated point. The new trajectory will have to ensure that it catches up with the time lost during obstacle avoidance in order to reach the final point at the specified time. Note that the new trajectory does not necessarily follow the initial generated trajectory as the new trajectory is based on the updated information.

4. Mobile Robot Platform

In this paper, the mobile robot used for the experimental works was converted from a standard remote control car as shown in Figure 3. The wheelbase length and width of the mobile are 174 mm and 191 mm, respectively. It has a similar structure to the normal car with front steering wheels and rear driving wheels. All four wheels have the same diameter which is 69 mm. The rear wheels are conventional fixed wheels on the rear axle and the front wheels are centred turning wheels on the front axle. The mobile robot is powered by a Ni-Cad battery and Pololu Robot Controller.

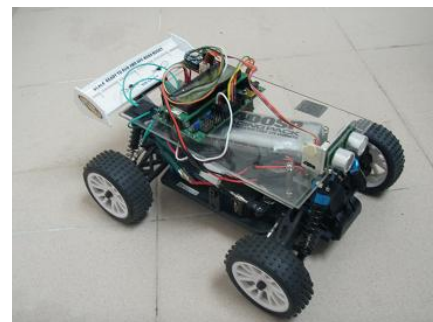


Figure 3. A car-like robot used in experimental works.

4.1 Calibration works

Prior to experimental works, the calibration works were conducted to establish two parameters which are:

- (i) PWM-steering angle relation
- (ii) PMW-velocity relation.

The related parameters for calibration work of steering angle are set as shown in

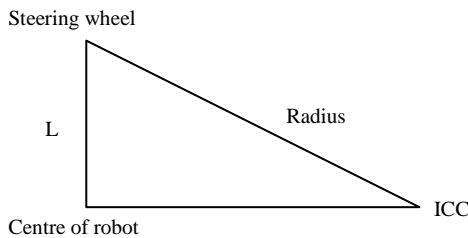


Figure 4. Calibration work for steering angle

From the results, the following linear relations were obtained by the Least Square Method:

$$\phi = 0.6502 \cdot PWM + 86.477, \text{ for turning left,} \quad (10)$$

$$\phi = -0.6717 \cdot PWM - 89.698, \text{ for turning right.} \quad (11)$$

For zero steering angle, the PWM value of 132 is used. Then the relation between PWM and steering angle is shown in

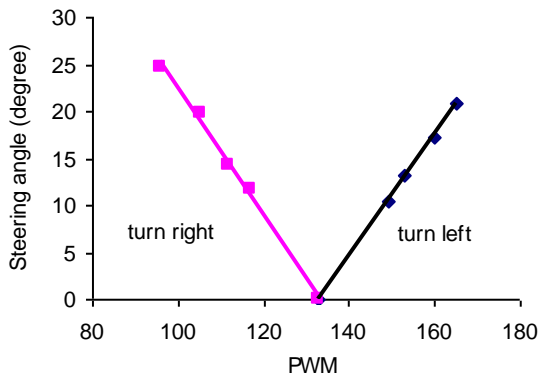


Figure 5.

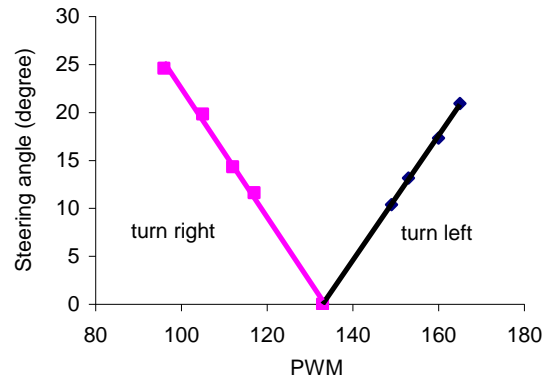


Figure 5. Relation between PWM values and steering angle.

The speed control is crucial to the time-critical motion planning as it can reflect the motion of the mobile robot. The speed is also controlled by the PWM value. The calibration work was conducted by taking time for 1000 m distance from start to final point for each PWM value as shown in Figure 6.

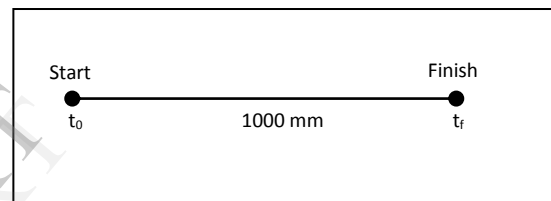


Figure 6. Calibration work for establishment of velocity

From the calibration work's results, with the Least Square Method, the following equation was established:

$$v = 2.038 \cdot PWM - 22.636. \quad (12)$$

The relation between the PWM and the speed is shown in Figure 7. Note that, the measured speeds were obtained during the battery is fully charged.

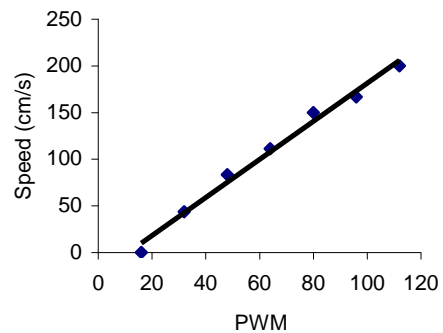


Figure 7. Relation between PWM values and speed.

Furthermore, in order to achieve remote control over the mobile robot, the Xbee RF module was used. The wireless connection strategy is shown in Figure 8. For this task, two Xbee RF modules were used as a Coordinator and a Router. Basically, the Coordinator was connected to the PC via USB and the Router was attached to the mobile robot via the General Purpose Input/Output (GPIO) port. Once the initial collision-free trajectory was generated, the data – output data – will be sent to the mobile robot by the Coordinator. Then the mobile robot will receive the data – as control input data – by the Router. These control inputs, which are steering angle and velocity, will be used to move the mobile robot. The communication can be a two-way communication with the Router will send the data and the Coordinator will receive the data. Such communication is necessary when the sensor's reading needs to be processed in the PC.

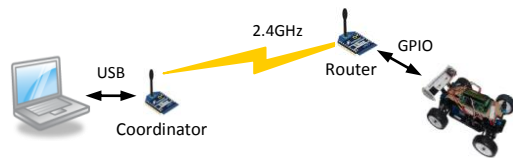


Figure 8. Wireless communication between operator and robot.

5. Simulation Architecture

The overall view of simulation architecture for the mobile robots' navigation is shown in Figure 9. The architecture is divided into three main levels: offline planning, online planning and obstacle data registration.

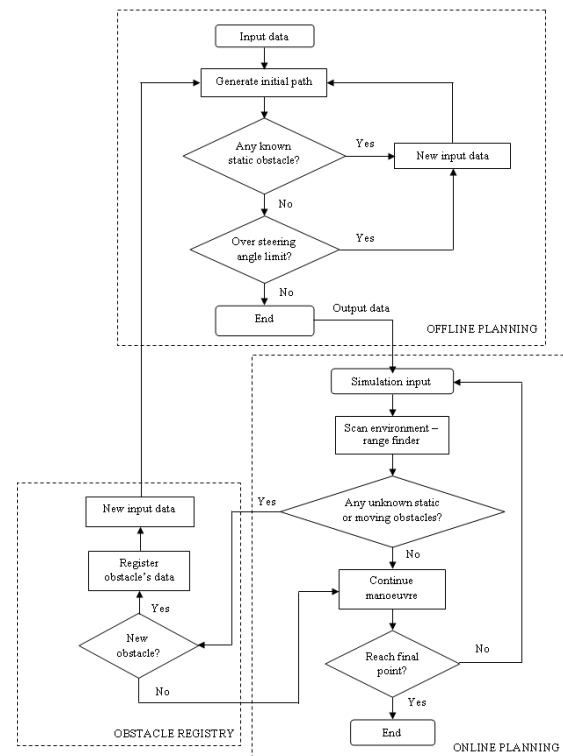


Figure 9. Simulation architecture for mobile robots

In the first level, the offline planner deals with a global initial trajectory generation. At this stage, only known static obstacles will be taken into consideration during trajectory generation from the initial point to the final point. The planner will first generate a pre-planned initial trajectory without considering known static obstacles. Then the planner will check whether there is any obstacle that is near or on the trajectory from the initial point to the final point. In the case where an obstacle is detected, the new input data at the detected obstacle's position, such as position, orientation, steering angle, velocity and time, will be obtained from the pre-planned initial trajectory. These input data will be used to determine a deviation point which is away from the obstacle. If there is more than one obstacle detected along the trajectory, the deviation point at each obstacle's position will be determined. Once the planner is finished checking the obstacles, a new global initial trajectory which is a collision-free trajectory will be generated from the initial point to the final point and passing all the deviation points. Furthermore, the steering angle limitation will also be considered at this stage.

Once the initial collision-free trajectory is generated, output data such as velocity and steering angle from the first level will be used as input data for the second level. In the second level, once the

mobile robot starts to navigate along the trajectory, the simulated laser scanner will also start to scan the environment. At this stage, the online planner deals with unknown static obstacles and also moving obstacles. If the mobile robot detects an obstacle which interferes with the robot's planned trajectory, the obstacle avoidance algorithm will be executed. The initial trajectory will be readjusted and a new trajectory will be generated based on the obstacle avoidance approach, which has been discussed earlier. This new trajectory will ensure the mobile robot avoids the obstacle and reaches the final point at the specified time.

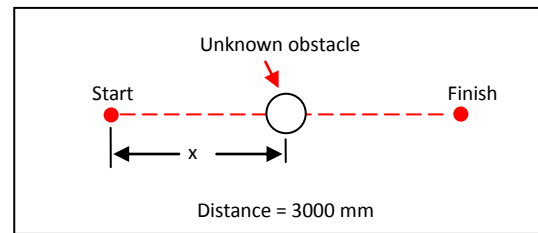
The third level deals with the obstacle data registration during online navigation. If the mobile robot detects an obstacle, the first detection will be registered into the system. Then when it detects another obstacle in the next step, the detected obstacle's position will be compared to the system. If the distance between the detected obstacle and the stored obstacle's position is smaller than obstacle's size, the new detected obstacle is assumed to be the same obstacle as the previously stored one. If the distance is larger than the obstacle's size, the detected obstacle will be considered as a new obstacle and this new obstacle's position will be registered into the system.

The algorithm will be processed in the PC using Matlab before transmitting the output data to the mobile robot. The data from sensors such as ultrasonic distance will then transmit to the PC to process at every second.

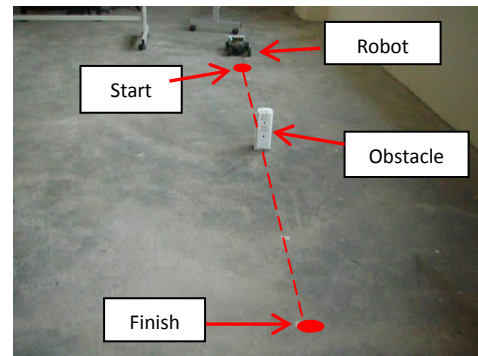
6. Results and Discussions

The experiment was further performed on the mobile robot in an unknown static environment. The purpose of this experiment is to validate the obstacle avoidance approach for the time-critical motion planning. In this experiment, an unknown static obstacle will be placed in the environment. The planner will generate the collision-free trajectory without the knowledge of unknown static obstacle and it is expected to detect and avoid the obstacle. Furthermore, the mobile robot is also expected to reach the final point at the desired time.

In the experiment, the mobile robot was required to move from the initial point to the final point as shown in Figure 10. The distance from the initial point to the final point is 3000 mm. An unknown static obstacle was placed randomly between the initial point and final point with the obstacle's diameter is 8 cm. The initial and final states of the mobile robot were $[0, 50, 0, 0, 0, 0]$ and $[300, 50, 0, 0, 0, 20]$, respectively.



(a)

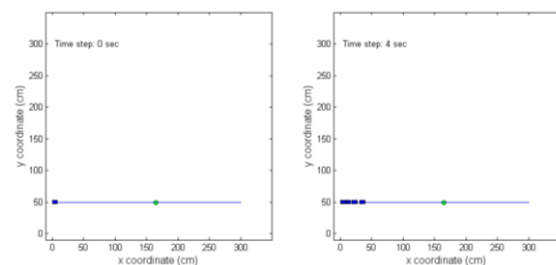


(b)

Figure 10. (a) Plan view (b) Actual experimental setup

The simulation results are shown in Figure 11 and Figure 12, respectively. Both simulation and experimental results can easily be compared through the respective figures. Using the initial and final states of the mobile robot, the planner generated an initial collision-free trajectory which is represented by a blue line as shown in Figure 11(a).

At the start of the experimental work, the mobile robot was followed the initial collision-free trajectory until it detected the obstacle in front of it as shown in Figure 12(c). Then the dynamic obstacle avoidance approach was executed and a new trajectory was generated from the point of detection to the final point. The mobile robot was then followed the new trajectory successfully until it reached the final point as shown in Figure 12(f).



(a) At time = 0 s

(b) At time = 4 s

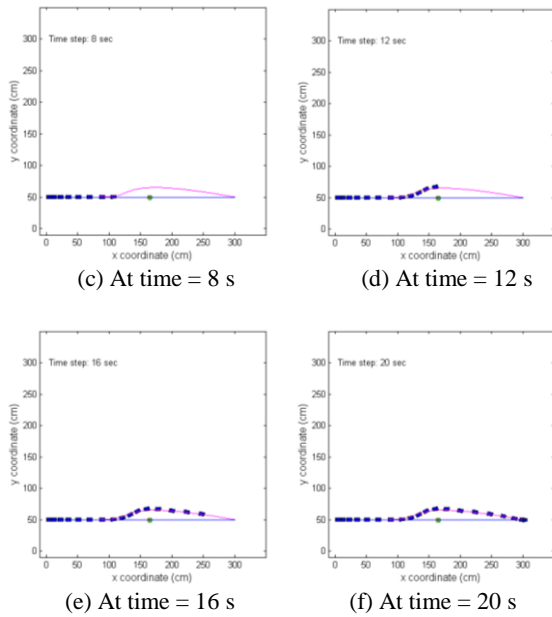


Figure 11. Mobile robot navigates in the unknown static environment (simulation)

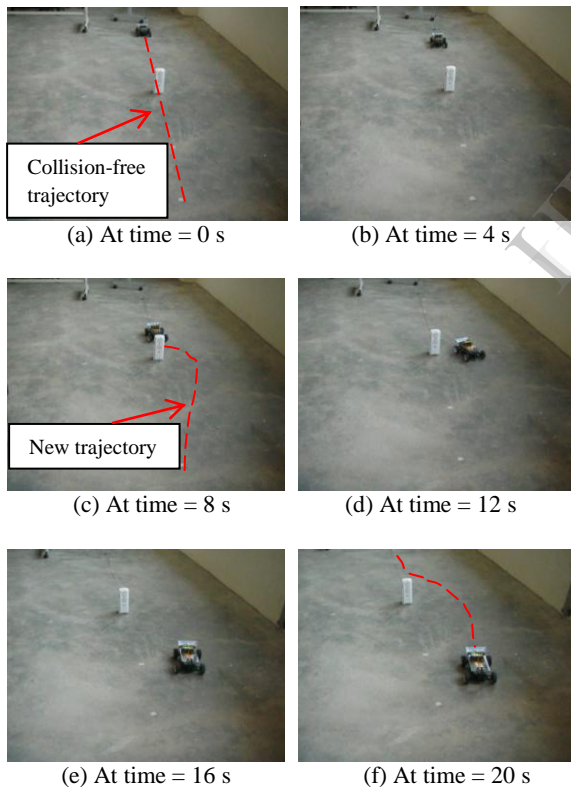


Figure 12. Mobile robot navigates in the unknown static environment (experiment)

The experiment results were then being compared with the theory as shown in Figure 13. During the experiment, the mobile robot detected the obstacle's location at (169, 50). The actual measured location of the unknown static obstacle was (163, 50). This was showed that the ultrasonic sensor was able to detect and locate the obstacle close to the actual location. In comparison with the theory, the mobile robot was able to follow the initial trajectory until it detected the obstacle and the new trajectory was generated in order to avoid the obstacle. From the point of detection, the mobile robot's movement was slightly deviated from its original trajectory and stopped before the final point. Despite errors in positions at the final point, the experiment has proven that the algorithms was practical to used in trajectory planning and control strategy for the mobile robot was translated the trajectory into robot control satisfactorily.

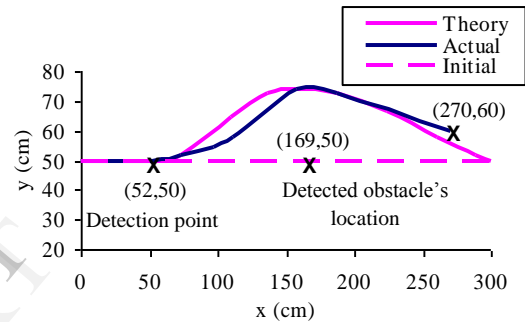


Figure 13. Theoretical and actual trajectory for Case 3

The experiment has been conducted in three trial runs and the errors in positions at the final point were compared with the theory as tabulated in Table 1. The maximum errors recorded in x -axis and y -axis are around 18.3% and 20%, respectively.

Table 1. Actual initial and final positions for Case 3

	x_s (cm)	y_s (cm)	x_f (cm)	y_f (cm)	x error (cm)	y error (cm)
Theory	0	50	300	50	-	-
Trial 1	0	50	270	60	-30	10
Trial 2	0	50	355	43	55	-7
Trial 3	0	50	350	40	50	-10

7. Conclusions

This paper presents a real-time control for a time-critical trajectory planning approach for a car-like robot in an unknown static environment. The mobile robot was adapted from a RC car. The mobile robot was controlled using wireless communication and the algorithm was processed in the PC before

transmitting the data to the mobile robot. Experimental works were conducted with the aim to investigate the effectiveness and validate the proposed algorithm in unknown static environments. From the results, the robot was able to follow the planned trajectory before it detected the obstacle. The mobile robot also was successfully detected and avoided the unknown obstacle before reaching the final point with acceptable position errors. This demonstrates the practicality and effectiveness of the proposed algorithm. And the performance of the algorithm is also satisfactory in term of position errors at the final point. In the future, the algorithm will further be tested for dynamic environments. Furthermore, a close-loop system with feedback control will be implemented to improve the performance of the algorithm.

References

- [1] Francois G. Pin and Hubert A. Vasseur, "Autonomous trajectory generation for mobile robots with non-holonomic and steering angle constraints", *IEEE International Workshop on Intelligent Motion Control*, Istanbul, Turkey, 20-22 August 1990.
- [2] A. Scheuer and Th. Fraichard, "Planning continuous-curvature paths for car-like robots", *IEEE/RSJ and Systems (IROS 2003)*, Las Vegas, USA, 27-31 October 2003.
- [9] K.G. Jolly, R. Sreerama Kumar and R. Vijayakumar, "A Bezier curve based path planning in a multi agent robot soccer system without violating the acceleration limits", *Robotics and Autonomous System*, vol. 57, 2009, pp. 23-33.
- [10] Bryan Nagy and Alonzo Kelly, "Trajectory generation for car-like robots using cubic curvature polynomials", *The 2001 International Conference on Field and Service Robots (FSR '01)*, Helsinki, Finland, 11 June 2001.
- [11] Simon Thompson and Satoshi Kagami, "Continuous curvature trajectory generation with obstacle avoidance for car-like robots", *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'05)*, Vienna, Austria, 28-30 November 2005.
- [12] Evangelos Papadopoulos and Ioannis Poulakakis, "Planning and obstacle avoidance for mobile robots", *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 21-26 May 2001.
- [13] F. Lamiroux and J.-P. Laumond, "Smooth motion planning for car-like vehicles", *IEEE Transactions on Robotics and Automation*, vol. 17, 2001, pp. 498-502.
- International Conference on Intelligent Robots and Systems (IROS'96)*, Osaka, Japan, 4-8 November 1996.
- [3] Soonshin Han, BoungSuk Choi and JangMyung Lee, "A precise curved motion planning for a differential driving mobile robot", *Mechatronics*, vol. 18, 2008, pp. 486-494.
- [4] M. Tounsi and J.F. Le Corre, "Trajectory generation for mobile robots", *Mathematics and Computers in Simulation*, vol. 41, 1996, pp. 367-376.
- [5] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots", *The 3rd International Symposium on Robotics Research*, Gourvieux, France, 1986, pp. 334-340.
- [6] Nicolas Montes, Alvaro Herraiez, Leopoldo Armesto and Josep Tornero, "Real-time clothoid approximation by rational Bezier curves", *IEEE International Conference on Robotics and Automation*, California, USA, 19-23 May 2008.
- [7] Ji-wung Choi, Renwick Curry and Gabriel Elkaim, "Path planning based on Bezier curve for autonomous ground vehicles", *Advances in Electrical and Electronic Engineering – IAENG Special Edition of the World Congress on Engineering and Computer Science (WCECS'08)*, San Francisco, CA, USA, 22-24 October 2008.
- [8] Jung-Hoon Hwang, Ronald C. Arkin and Dong-Soo Kwon, "Mobile robots at your fingertips: Bezier curve on-line trajectory generation for supervisory control", *IEEE/RSJ International Conference on Intelligent Robots*
- [14] Yongji Wang, "Nonholonomic motion planning: a polynomial fitting approach", *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 22-28 April 1996.
- [15] Zhihua Qu, Jing Wang and Clinton E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles", *IEEE Transaction on Robotics*, vol. 20, 2004, pp. 978-993.
- [16] Yi Guo, Zhihua Qu and Jing Wang, "A new performance-based motion planner for nonholonomic mobile robots", *The 3rd Performance Metrics for Intelligent Systems Workshop*, Gaithersburg, MD, USA, 16-18 September 2003.
- [17] Yi Guo, Yi Long and Weihua Sheng, "Global trajectory generation for nonholonomic robots in dynamic environments", *2007 IEEE International Conference on Robotics and Automation*, Roma, Italy, 10-14 April 2007.
- [18] Yi Guo and Tang Tang, "Optimal trajectory generation for nonholonomic robots in dynamic environments", *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 19-23 May 2008.
- [19] M.Sani M.Hashim and Tien-Fu Lu, "Time-dependent motion planning for nonholonomic mobile robots", *In Proceedings of the 9th International IFAC Symposium on Robot Control (SYROCO'09)*, Gifu, Japan, 9-12 September 2009.

[20] Sani Hashim and Tien-Fu Lu, "A new strategy in dynamic time-dependent motion planning for nonholonomic mobile robots", *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO'09)*, Guilin, China, 18-22 December 2009.

[21] M.Sani M.Hashim and Tien-Fu Lu, "Multiple waypoints trajectory planning with specific position, orientation, velocity and time using geometric approach for a car-like robot", *2009 Australasia Conference on Robotics and Automation (ACRA'09)*, Sydney, Australia, 2-4 December 2009.

[22] Wenjie Dong and Yi Guo, "New trajectory generation methods for nonholonomic mobile robots", *Proceedings of the 2005 International Symposium on*

Collaborative Technologies and Systems, 2005, pp. 353-358.

[23] K.Jiang, L.D. Senevirate and S.W.E. Earles, "Time-optimal smooth-path planning for a mobile robot with the kinematic constraints", *Robotica*, vol. 15, 1997, pp. 547-553.

[24] Nirmal Baran Hui, V. Mahendar and Dilip Kumar Pratihari, "Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approach", *Fuzzy Sets and Systems*, vol. 157, 2006, pp. 2171-2204.

IJERT