2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE 2012), December 3-4, 2012, Kota Kinabalu Malaysia

Dynamic Obstacle Avoidance Approach for Car-like Robots in Dynamic Environments

Mohd Sani Mohamad Hashim,
School of Mechtronic Engineering
Universiti Malaysia Perlis (UniMAP),
Pauh Putra Campus, 02600 Arau, Perlis, MALAYSIA.
sanihashim@unimap.edu.my

Tien-Fu Lu,
School of Mechanical Engineering,
The University of Adelaide, Adelaide,
5005 South Australia, AUSTRALIA.
tien-fu.lu@adelaide.edu.au

Hassrizal Hassan Basri
School of Mechtronic Engineering
Universiti Malaysia Perlis (UniMAP),
Pauh Putra Campus, 02600 Arau, Perlis, MALAYSIA.
hassrizal@unimap.edu.my

Abstract—**In this paper, a new dynamic obstacle avoidance approach for nonholonomic mobile robots in dynamic environments is presented. In dynamic environments, the mobile robot is expected to encounter and safely avoid the obstacles along its way. This inevitably will delay the mobile robot in keeping to its original planned timeframe. To address this scenario, the proposed approach will ensure the mobile robot is able to gain the time lost during obstacle avoidance and reach the final point at the specified time. This approach is based on the dynamic trajectory planning scheme which utilized the replanning approach in order to avoid the obstacle. The performance of the proposed approach is tested through simulations in a simplified city-like dynamic environment.**

Keywords-nonholonomic robot; car-like robot; trajectory planning; obstacle avoidance; dynamic environment

1. INTRODUCTION

Avoiding obstacles is one of the problems for a mobile robot to navigate in static and dynamic environments. In dynamic environment, where there are static and moving obstacles, the task becomes more complicated and difficult in comparison to static environments. Therefore, many approaches have been introduced in previous research in order to develop an effective and reliable obstacle avoidance capability for mobile robots to navigate in static and dynamic environments.

Fajen and Warren [1] introduced a new solution for obstacle avoidance based on observing the human behaviour in dynamic environments. In their paper, the aim is to apply the dynamic model to the robot behaviour of steering towards a goal and avoiding the obstacles. Once the set of behaviour variables for steering and obstacle avoidance have been identified, the general form of the model will be introduced. This work has been extended by Fajen *et al.* [2] by using visually-guided locomotion in a dynamic environment in order to identify a set of behavioural variables for steering and obstacle avoidance.

The most commonly used method for solving the obstacle avoidance problem is based on the potential field method, firstly proposed by Khatib [3]. Then Huang *et al.* [4] proposed a vision-guided navigation approach by adapting Fajen and Warren's work on human behaviour navigation and this approach was expressed as a potential field. In their study, the potential field is used to control the angular acceleration and heading of the robot in order to steer it toward the goals and to avoid the obstacles during robot navigation. However, this approach has a limitation since they used angular width of the obstacle rather than distance, yet a large obstacle can also has the same angular width as a smaller obstacle.

Furthermore, Hamner *et al.* [5] also proposed an extension method based on Fajen and Warren formulation. The proposed method can learn the parameters of the control model automatically by observing behaviour of the human driver. In addition, Hamner *et al.* introduced a speed control function based on the obstacle's distance and angle in their method. This speed control function slows down the vehicle as the obstacles get closer, which gives time to the vehicle to turn and avoid the obstacles. However this method also allows a sharp turning which has a negative impact for the vehicle motion. Moreover, their results showed that the vehicle attempts to follow a far path while avoiding large obstacles and gave conservative results.

In 2008, Jolly *et al.* [6] proposed a method for avoiding the dynamic obstacle by modifying the initial generated Bezier curve. At the initial stage, the robot will travel along the original curve. Once the obstacle has been detected, the new modified Bezier curve will be generated and the new trajectory will ensure the obstacle is avoided. However, in their paper, the approach was tested only for holonomic mobile robot and not for a car-like robot.

Therefore, in this paper a new dynamic obstacle avoidance approach is introduced in order to fulfill the requirement of a time-critical trajectory planning which has been introduced by the authors in previous publication [7]. This obstacle avoidance approach will ensure the mobile robot will reach the final point at the specified time and gain the time lost during avoiding the obstacle. This approach will be useful for task-based applications such as patrolling a large area and robot soccer, which require the mobile robot to be at the specified point at the specified time with desired orientation.

2. KINEMATICS OF A NONHOLONOMIC ROBOT

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. The robot state in Cartesian space can be represented by $q=[x, y, \theta, \o, v, t]^T$, 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, $\o$ 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.

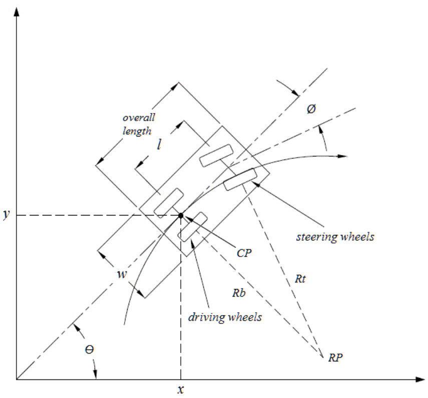


Figure 1. A nonholonomic mobile robot

The following are some general assumptions made for this work:

- The mobile robot moves on a planar surface,
- The wheels rotate without slipping,
- The wheels are not deformable,
- 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 $-\o_{max} \leq \o \leq \o_{max}$.

3. Dynamic Obstacle Avoidance Approach

In this study, the obstacle avoidance approach deals with both static and moving obstacles in a 2D workspace. The approach is based on a dynamic trajectory planning scheme [6], In a dynamic trajectory planning scheme, the mobile robot will replan and modify its trajectory once it detects an obstacle and the newly generated trajectory may differ from the initially planned trajectory. However, instead of using the Bezier curves, which were used by Jolly *et al*. [6], polynomial curves have been adopted in this study. The reason behind this is to ensure that the mobile robot will pass through all the control points to have a better control for the mobile robot's motion, compared to the Bezier curves, which only pass through the first and last control points [6]. Furthermore, the dynamic trajectory planning scheme is divided into two planning schemes, which are utilised to avoid static obstacles and moving obstacles.

A. Avoiding static obstacles

In this study, the static obstacles are divided into two categories: known and unknown. Known static obstacles are known in advance to the planner during offline planning, while unknown static obstacles are unknown to the planner and will only be detected by the sensor during navigation. For the known static obstacles, the planner will consider them in the initial stage during generating the trajectory. Thus the generated trajectory should be away from the potentially colliding obstacles. Meanwhile, the unknown static obstacles will only be considered when the mobile robot starts to navigate through the environment. The general view of avoiding an unknown static obstacle is illustrated in Figure 2.

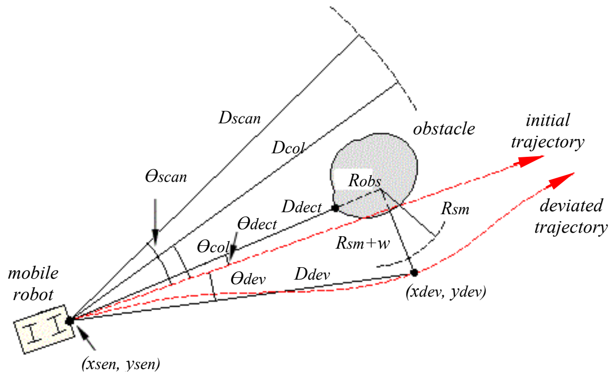


Figure 2. Avoiding an unknown static obstacle

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:

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

$$\theta_{dev} = \theta_{dect} + \tan^{-1}\left(\frac{R_{sm} + w}{D_{dect} + R_{obs}}\right) \quad (3)$$

$$x_{dev} = x_{sen} + d_{dev} * \cos\theta_{dev} \quad (4)$$

$$y_{dev} = y_{sen} + d_{dev} * \sin\theta_{dev} \quad (5)$$

Once the deviated point is obtained, a new replanned trajectory (deviated trajectory) is generated from the current point to the final point, through the deviated point. This replanned 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 replanned trajectory does not necessarily follow the initial generated trajectory as the replanned trajectory is based on the updated actual information from the mobile robot.

B. Avoiding moving obstacles

The strategy to avoid a moving obstacle is usually based on prior information of the moving obstacle's velocity [8,9]. However, in this study the strategy is based on the direction and position of the moving obstacle, which is unknown to the planner and at this stage, the size of the moving is known in prior. Furthermore, the moving obstacle's direction will influence the selection of appropriate strategy to avoid it. For instance, if the moving obstacle is approaching perpendicularly to the mobile robot, the mobile robot will avoid the moving obstacle as illustrated in Figure 3(a). On the other hand, if the moving obstacle is approaching from the opposite direction of the mobile robot, the moving obstacle is treated as a static obstacle and the mobile robot will avoid the obstacle as illustrated in Figure 3(b).

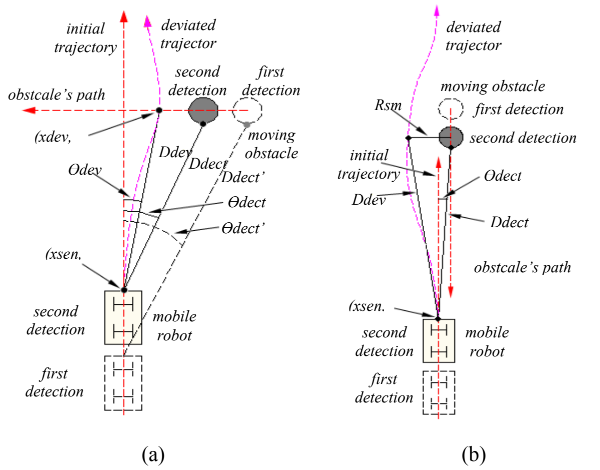


Figure 3. Avoiding a moving obstacle (a) perpendicular direction to the mobile robot and (b) in opposition to the mobile robot.

Despite the direction of the moving obstacle, the mobile robot will predict the possibility of collision between the mobile robot and the moving obstacle. As shown in Figure 4(b), when the mobile robot first detects a moving obstacle, the position for both the mobile robot and the moving obstacle will be registered into the registry. Then, when the next detection occurs, the system will compare the stored position (first detection) with the current position (second detection) to obtain direction and distance between these two locations for both the mobile robot and the moving obstacle. In addition, the planner will estimate the speed of the moving obstacle. From this information, the system can predict the mobile robot's and moving obstacle's position for the next two steps. If the predicted moving obstacle's position falls inside the collision radius of the mobile robot, then the collision is likely to happen as shown in Figure 4(c). The collision point (x_{col}, y_{col}) and the deviation point (x_{dev}, y_{dev}) are then be determined by using the following equations:

$$x_{col} = x_{sen} + D_{dect} * \cos\theta_{dect} \tag{6}$$

$$y_{col} = y_{sen} + D_{dect} * \cos\theta_{dect} \tag{7}$$

$$D_{col} = \sqrt{(x_{movobs} - x_{col})^2 + (y_{movobs} - y_{col})^2} \tag{8}$$

$$x_{dev} = x_{col} - (r_{movobs} + w) \tag{9}$$

$$y_{dev} = y_{col} \tag{10}$$

where,

r_{movobs} = size of a moving obstacle.

Then a new trajectory (deviated trajectory) will be generated from the mobile robot's current point to the final point, through the deviation point as shown in Figure 4(d). Note that the position of the moving obstacle at every time step is prior knowledge to the system, and the moving obstacle is assumed to follow its path precisely and does not deviate from its originally planned path.

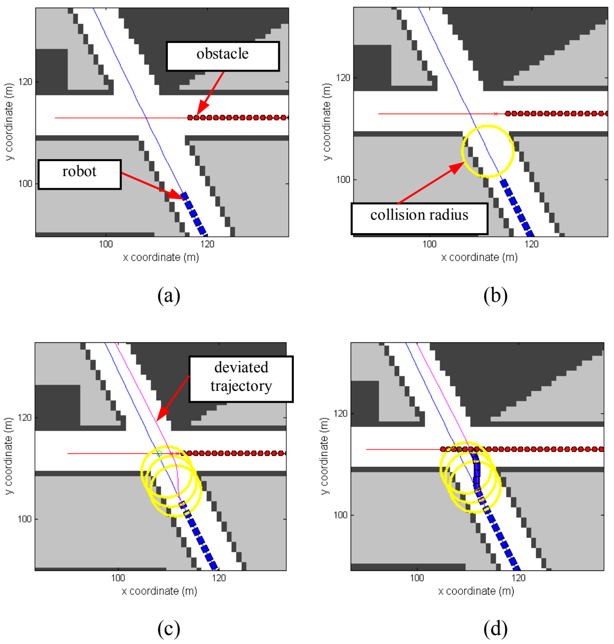


Figure 4. Collision prediction approach (a) before detection of the obstacle, (b) first detection, (c) predicted position falls inside the collision radius, and (d) obstacle avoidance approach implemented.

4. Simulation Results and Discussions

In this section, the simulations are based on the multiple waypoints trajectory planning in a city-like environment as shown in Figure 5. The simulated city-like environment consists of the combination of static and dynamic obstacles, which makes the trajectory planning more complicated. Furthermore, the planner needs to ensure the mobile robot is within the road and shall not move outside the road's line. The simulation work was conducted in Matlab.

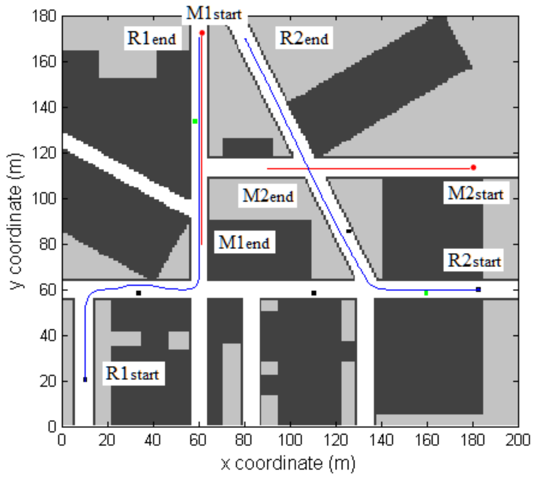


Figure 5. Initial trajectories in a city-like map.

All the parameters used for the mobile robots; R1 and R2, are listed in Table I and Table II, respectively. North direction of the map is set pointing up on the map. As listed in Table I, R1 starts from the bottom of the map at point (10, 20) and facing north. At the first junction, it needs to turn right. The first and second waypoints are set to ensure the mobile robot can turn at the junction smoothly. Then, it needs to move along the road until it reaches the second junction. It then needs to turn left and move until it reaches the final point (60, 170) at the 120th second with 90° orientation.

TABLE I. Parameters for the First Mobile Robot (R1)

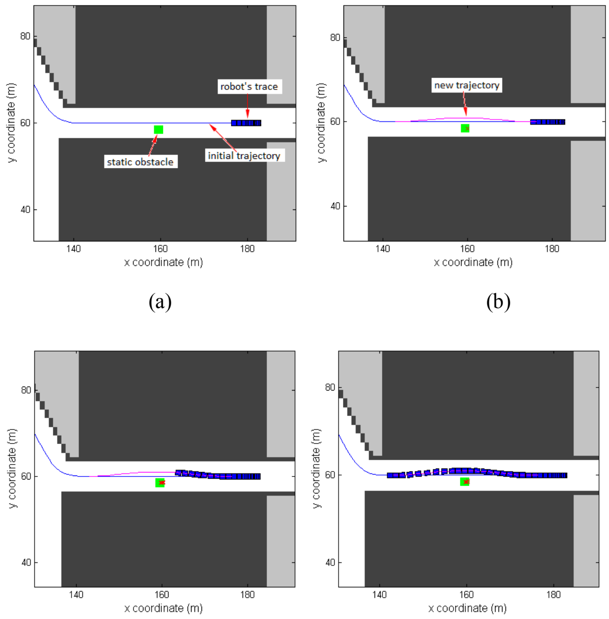
Points	t (sec)	x (m)	y (m)	θ (°)	ϕ (°)	v (m/s)
1	0	10	20	90	0	0
2	20	10	50	90	0	1
3	30	20	60	0	0	2
4	50	50	60	0	0	1
5	60	60	67	90	0	2
6	120	60	170	90	0	0

As listed in Table II, R2 starts from the right side of the map at point (183,60) and facing west. Then at the junction, it needs to turn right and move along the road until it reaches the final point (80,170) at the 120th second with 60° orientation. Note that the road is tilted at about 60° from x-axis.

TABLE II. Parameters for the Second Mobile Robot (R2)

Points	t (sec)	x (m)	y (m)	θ (°)	ϕ (°)	v (m/s)
1	0	183	60	0	0	0
2	30	143	60	0	0	1
3	40	133	65	-60	0	1
4	120	80	170	-60	0	0

In addition, there are two moving obstacles in the map as shown in Figure 5. The first moving obstacle (M1) starts from the north of the map and moves straight down to the south of the map. The initial and final point for M1 is (61, 180) and (61, 80), respectively. The second moving obstacle starts from east of the map and finishes at the middle of the map. The initial and final point for M2 is (180, 113) and (90, 113), respectively. Both moving obstacles move from their respective initial points and reach their final points at the 100th second.



(a) (b) (c) (d)

Figure 6. (a) Before detecting an obstacle. (b) Obstacle detected at the 9th second. (c) Starts to move along new trajectory. (d) Reaches the first waypoint at the 30th second.

The initial trajectories for mobile robots and moving obstacles are shown in Figure 5. Once all the trajectories were generated, the mobile robots and the moving obstacles were started to move along their respective trajectories. At the 9th second, R2 detected an unknown static obstacle as shown in Figure 6 (b). Then a new trajectory was generated from the detection point to the closest waypoint, which was in this case the first waypoint. R2 started to move along the new trajectory as shown in Figure 6(c) and reached the first waypoint at the 30th second as shown in Figure 6(d).

Furthermore, at the 67th second, R2 detected a moving obstacle (M2) coming from the right side of it as shown in Figure 7(b). It then predicts whether it might collide with the moving obstacle or not. In this case, collision is expected to happen and a new trajectory is generated from detection point to the closest waypoint, which is the final point, based on the obstacle avoidance algorithm of a moving obstacle. Then R2 started to move along the new trajectory as shown in Figure 7(c). Also as we can see, the mobile robot actually slowed down to cautiously passing through the moving obstacle as shown in Figure 7(d).

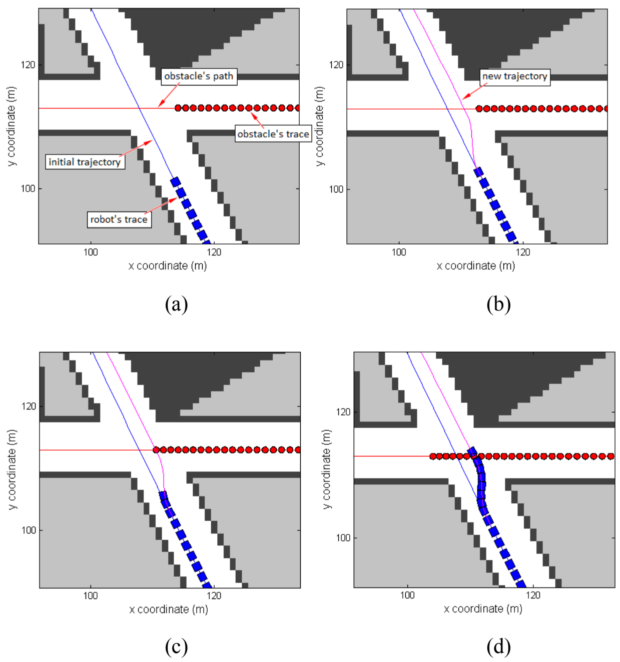


Figure 7. (a) Before detecting an obstacle. (b) Obstacle detected at the 67th sec. (c) Starts to move along new trajectory. (d) Passes through moving obstacle safely.

As we can see in Figure 5, the initial trajectory for R1 had already considered known static obstacles during offline planning. Then R1 started to move along the initial trajectory, passing through all the waypoints. However at the 68th second, R1 detected a moving obstacle (M1) as shown in Figure 8(b). Also R1 checked the direction of moving obstacle and in this case, M1 came from the opposite direction of R1. Therefore, M1 was treated as a static obstacle and a new trajectory was generated from the current point to the final point, through the deviation point. Then R1 started to move along the new trajectory and safely avoided M1 as shown in Figure 8(c) and (d). Furthermore, after avoiding the moving obstacle, R1 detected an unknown static obstacle at the 86th second and successfully avoided it.

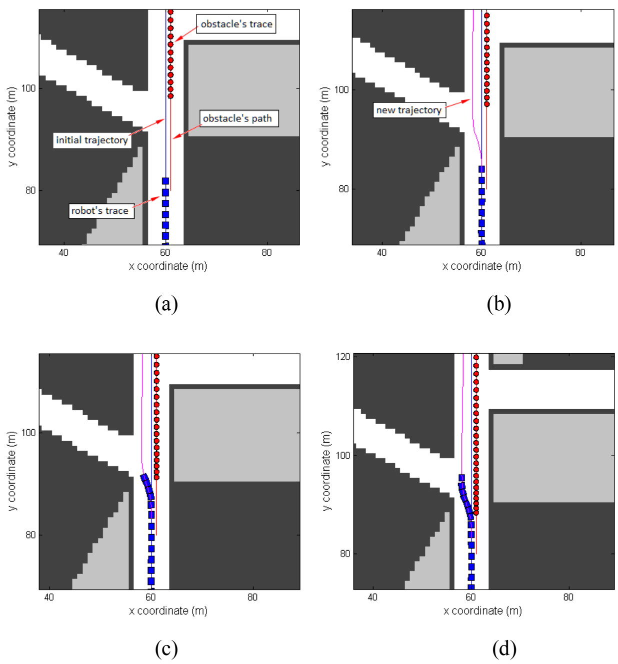


Figure 8. (a) Before detecting an obstacle. (b) Obstacle detected at the 68th sec. (c) Starts to move along new trajectory. (d) Passes through moving obstacle safely.

Figure 9 shows the final overall simulation results at the 120th second. As we can see, both mobile robots reached the final point at the specified time, position and orientation with certain errors as shown in Table III. As the algorithm tried to replicate the actual movement of a car, the errors came from the estimation for the actual mobile robot's position using the following equations:

$$d = v * \Delta t + 0.5 * (v_{next} - v) * \Delta t \tag{18}$$

$$\Delta\theta = 2 * \sin^{-1}\left(\frac{d \tan\phi}{2 * l}\right) \tag{19}$$

$$\theta_{new} = \theta_{old} + \Delta\theta \tag{20}$$

$$x_{new} = x_{old} + d * \cos\theta_{new} \tag{21}$$

$$y_{new} = y_{old} + d * \sin\theta_{new} \tag{22}$$

where

d = distance from the current point to the next point.

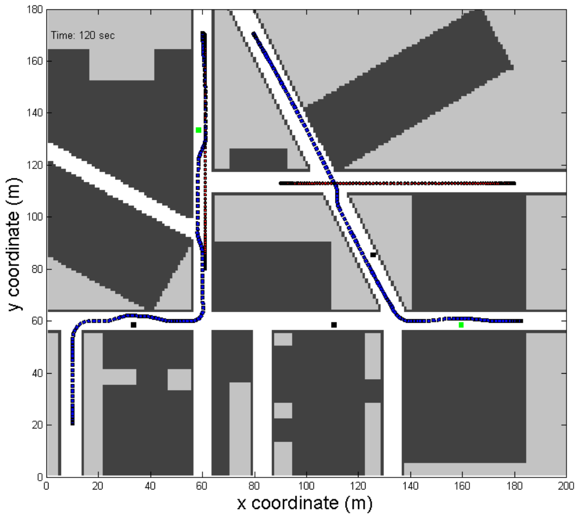


Figure 9. Final result at the 120th second.

From Table III, the errors are reasonably small as a result of the online planning approach. At every time step, the online planner will use the actual data to obtain as close as possible to the next pre-planned position of the mobile robot. This means the planner will need to determine a new steering angle using the actual position and orientation, and the pre-planned velocity of the mobile robot. This practice will eliminate or at least reduce the errors at every time step. Furthermore, the mobile robots successfully passed through all the waypoints and avoided all the static and moving obstacles.

TABLE III. ERRORS OF MOBILE ROBOTS' POSITIONS AT THE FINAL POINT.

	Actual			Error		
	x (m)	y (m)	θ (°)	x (m)	y (m)	θ (°)
Robot 1	59.999	169.99	89.427	0.001	0.01	0.573
Robot 2	79.972	170.04	-62	0.028	0.04	2

5. CONCLUSIONS AND FUTURE WORKS

This paper presents a new dynamic obstacle avoidance approach for nonholonomic mobile robots in dynamic environments, with a presence of static and dynamic obstacles. This approach was adopted in the time-critical trajectory planning that requires the mobile robot to reach the final point at the specified time with the desired orientation. This dynamic obstacle avoidance approach ensures the mobile robot will be able to gain the time lost during obstacle avoidance and reach the final point at the specified time. The presented obstacle avoidance approach has been implemented and investigated in simulation works. From the results, all the mobile robots are able to reach the final point at the specified time with certain reasonable position errors. In the future, there are areas that will be studied further such as trajectory optimization and multiple robots coordination. Furthermore, this approach will be verified through experimental works using an actual car-like robot.

ACKNOWLEDGMENT

This paper is funded by UniMAP Short Term Grant (STG) number 9001-00353. Furthermore, we would like to thank every single person who involved directly or indirectly in this paper.

REFERENCES

[1] Fajen, B. R. and W. H. Warren, "Behavioral dynamics of steering obstacle avoidance and route selection", Journal of Experimental Psychology: Human and Perception and Performance, vol.29(2), pp. 343-362, 2003.

[2] Fajen, B. R., W. H. Warren, S. Temizer and L. P. Kaelbling, "A dynamic model of visually-guided steering, obstacle avoidance and route selection", International Journal of Computer Vision, vol.54(1/2/3), pp. 13-34, 2003.

[3] Khatib, O., "Real-time obstacle avoidance for robot manipulator and mobile robots", International Journal of Robotics Research, vol.5(1), pp. 90-98, 1986.

[4] Huang, W. H., B. R. Fajen, J. R. Fink and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function", Robotics and Autonomous Systems, vol.54, pp. 288-299, 2006.

[5] Hamner, B., S. Singh and S. Scherer, "Learning obstacle avoidance parameters from operator behavior", Journal of Field Robotics, vol.23(11/12), pp. 1037-1058, 2006.

[6] 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, pp. 23-33, 2009.

[7] 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.

[8] 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.

[9] 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, pp. 978-993, 2004.