Australasian Conference on Robotics and Automation (ACRA), December 2-4, 2009, Sydney, Australia

Multiple Waypoints Trajectory Planning with Specific Position, Orientation,Velocity and Time using Geometric Approach for A Car-like Robot

Mohd Sani Mohamad Hashim
School of Mechanical Engineering
The University of Adelaide
mohd.mohamadhashim@adelaide.edu.au
Tien-Fu Lu
School of Mechanical Engineering
The University of Adelaide
tien-fu.lu@adelaide.edu.au

Abstract

This paper proposes a new approach to plan the motion for a car-like mobile robot navigating in static environments. A multiple waypoints trajectory planning is introduced in order to ensure the planned trajectory is smooth and is able to guide the robot to pass through all the required waypoints at the specified position, orientation, velocity and time. This presented approach uses cubic and quintic polynomials to obtain a smooth trajectory. Furthermore, the generated trajectory is time-depended and the kinematic constraints of the mobile robot are taken into account during the generation of trajectory. While navigating, the mobile robot is able to negotiate with unexpected static obstacles, adjust its original trajectory and travel speed, and finally reach the final location at the specified time, orientation and velocity. The simulation results demonstrate the effectiveness and practicality of the proposed approach.

1 Introduction

Path planning for car-like robots has been widely studied in the past few decades. In many applications, the global path planning is required at the initial stage and prior to robot's navigation, in order to obtain a smooth and safe path. Generally the focus of path planning is to obtain the optimal path such as the shortest path, fastest path and/or minimum-time path. However in certain situation, the mobile robot is not only require to arrive at the specified point with correct orientation, but also at the specified time. This situation may have a significant ramification for applications to which a mobile robot is tasked for such as large areas patrol, goods delivery and multi-robot coordination, such as soccer robots. Hence trajectory planning strategy is crucial in order to meet the requirements of the orientation, position as well as the timing. In addition, a mobile robot is governed by kinematic constraints such as the steering angle limitation especially for a car-like robot. Correct strategy will ensure the mobile robot to navigate in the environment safely and smoothly. Furthermore, a multi waypoint navigation method is needed where the control of position, orientation and velocity are required at every specified points, which cannot be satisfied by a single trajectory planning. For example, to patrol large areas, besides the requirement of reaching the checkpoint at the specified time, the mobile robot may also need to arrive at certain orientation to cover the angle of view at each checkpoints, in order to improve the safety and coverage of the area.

In [Guo *et al.*, 2003], they presented a global trajectory planning, in which the piecewise constant parameterization is used to generate the feasible trajectory. The trajectory consists of several waypoints and if the mobile robot detects an obstacle, the trajectory will be modified and the waypoints are maintained. Therefore the mobile robot will be able to avoid the obstacle. However in their approach, there are a few issues such as the orientation at every waypoint is compromised, in which the orientation can be changed to satisfy the new trajectory, in order to avoid the obstacle and also the discontinuity problem at the waypoints that was caused by global axis rotational. Then in 2007, [Guo *et al.*, 2007] has improved the approach. From the simulation results, the discontinuity problem has been solved and a smooth trajectory has been achieved. However the robot's orientation at every waypoint still cannot be controlled and has to be compromised.

So far, there is no extensive study on developing a multiple waypoint trajectory planning with specific parameters control including velocity and time, at every waypoint. Therefore, the aim of this study is to develop an effective multiple waypoint trajectory planning method for a car-like robot with position, orientation, velocity and time control at every waypoint to overcome the aforementioned limitations. The development of the motion planning method is based on the geometric approach proposed by [Dong and Guo, 2005] and then extended by the authors [Hashim and Lu, 2009]. Time-dependent motion planning is used to generate trajectories for multiple waypoints trajectory planning in order to achieve the aim of this study.

2. Trajectory generation using geometric approach

Geometric approach has been receiving great interests in path planning for nonholonomic systems as this approach can provide a smooth and safe path. Basically the geometric approach generates the path by assembling curves to connect all the points. There were many different types of curves have been studied, such as clothoids [Kanayama and Miyake, 1986], Bezier curves [Jolly *et al*., 2009] and polynomial curves [Kelly, 2003; Nagy and Kelly, 2001; Dong and Guo, 2005]. However, besides polynomial curves, most of these curves are based on approximation which means curves are not necessary passing through all the required points. Hence, using polynomial curves are the best solution as they can also provide a smooth path if passing through all the required points is necessary.

Therefore, the algorithm for time-dependent motion planning in this presented study is developed based on geometric approach using polynomial curves. The assumptions at this stage are:

- The desired initial and final state are known which include Cartesian-coordinate position, orientation, steering angle, velocity and time.
- The environment's terrain is considered flat.
- The steering wheels of the mobile robot are acted as a single steering wheel.

A geometry model of a car-like mobile robot is shown in Figure 1, where the front wheels are the steering wheels and the rear wheels are the driving wheels. The centre point of the mobile robot (CP) is located at the middle of the rear axle. Given the generalized state is $q=[x,y,\theta,\phi,v,t]^T$, with ($x$, y) are the Cartesian coordinate, θ is the orientation of mobile robot with respect to the x-axis in Cartesian coordinate and ϕ is steering angle.

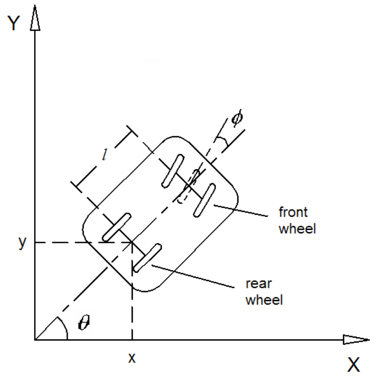


Figure 1 A car-like mobile robot.

Therefore, the state space that represents the kinematic constraints of this mobile robot can be obtained from:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\\ \dot{\phi}\end{pmatrix}=\begin{pmatrix}\cos\theta\\ \sin\theta\\ \tan\phi/l\\ 0\end{pmatrix}\rho u_1+\begin{pmatrix}0\\0\\0\\1\end{pmatrix}u_2 \qquad (1)$$

Where,
ρ = radius of rear wheel.
u_1 = angular velocity of the driving (rear) wheel.
u_2 = steering velocity of steering (front) wheel.

From (1), we have:

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

The boundary conditions for the mobile robot have been set as follows:

$$\begin{aligned}q(t_o)&=q^0=[x_0,y_0,\theta_0,\phi_0,v_0,t_0]^T,\\ q(t_f)&=q^f=[x_f,y_f,\theta_f,\phi_f,v_f,t_f]^T,\end{aligned} \qquad (3)$$

With the consideration of the boundary conditions, cubic and quintic polynomials have been adopted for the position equations of the mobile robot as:

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

Then the orientation and velocity equation can be derived as follows:

$$\theta=\tan^{-1}\left(\frac{b_1+2b_2t+3b_3t^2+4b_4t^3+5b_5t^4}{a_1+2a_2t+3a_3t^2}\right) \qquad (5)$$

$$\begin{aligned}v=\rho u_1&=(a_1+2a_2t+3a_3t^2)\cos\theta+\ldots\\ &(b_1+2b_2t+3b_3t^2+4b_4t^3+5b_5t^4)\sin\theta\end{aligned} \qquad (6)$$

The range of θ has to be within $\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ to avoid the singularity. If the θ is out of the range, the global axis will be rotated and the algorithm is executed in the new global axis. The development of these equations has been discussed in detail by [Hashim and Lu, 2009].

In order to obtain the steering angle accurately, a real car-like robot model was developed. The model is based on an all-terrain vehicle (ATV), which is known as Quadbike and it is assumed to be acted as a tricycle in this study, which the rear wheels are the driving wheels and the front wheel is a steering wheel. The geometry model of the Quadbike is modelled based on the works by [Liddy and Lu, 2007] and [Hashim and Lu, 2009] as shown in Figure 2.

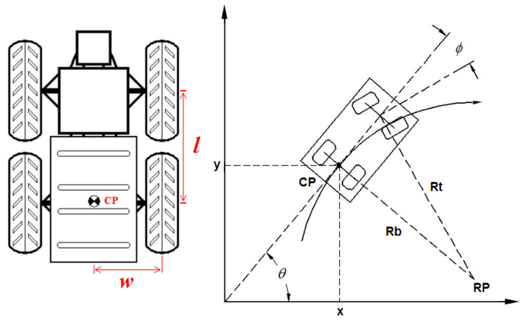


Figure 2 Geometry model of a mobile robot [Hashim and Lu, 2009].

In this study, the centre point of the Quadbike (CP) is defined to be located at the middle of the rear axle instead of the centre of the Quadbike in order to match the geometry model in Figure 1. The inputs for this simulation environment are orientation (θ) and velocity (v). To find the steering angle (ϕ):

$$d = u * \Delta t + \frac{1}{2} \Delta v * \Delta t \qquad (7)$$

$$Rb = \frac{d}{2 * \sin^{-1}(\Delta\theta / 2)} \qquad (8)$$

$$\phi = \tan^{-1}\left(\frac{l}{Rb}\right) \qquad (9)$$

To find position x and y, the equations can be reversed by using steering angle and velocity as inputs, as shown by the following equations:

$$Rb = \frac{l}{\tan\phi} \qquad (10)$$

$$\Delta\theta = 2 * \sin^{-1}\left(\frac{d/2}{Rb}\right) \qquad (11)$$

$$\theta_{new} = \theta_{old} + \Delta\theta \qquad (12)$$

$$x_{new} = x_{old} + d * \cos\theta \qquad (13)$$

$$y_{new} = y_{old} + d * \sin\theta \qquad (14)$$

3. Obstacle avoidance approach

Obstacle avoidance approach in the presented work deals with static obstacles. The obstacle avoidance approach used in this study is a dynamic trajectory planning scheme. This approach is based on the work done by [Jolly *et al.*, 2009] and [Hashim and Lu, 2009]. In a dynamic trajectory planning scheme, the mobile robot will replan and modify its trajectory once it detects an obstacle and the new trajectory will differ from the initially planned trajectory.

The steps of the dynamic trajectory planning scheme is shown by the flowchart in Figure 3. Once the initial state and the final state have been set, an initial trajectory will be generated in consideration of known static obstacles. Then, while the mobile robot navigates along the initial trajectory, the mobile robot will also scan and detect the presence of obstacles using modelled laser range finder. If obstacle is detected, robot data, such as position, orientation, steering angle, velocity and time, at the current position will be obtained and will be used to avoid the obstacle. The algorithms will then check the distance between the current position and the detected obstacle. The purpose of checking the distance is to ensure the mobile robot will be able to catch-up the time delayed due to avoiding the obstacle. Then the algorithm will estimate the time needed to avoid the obstacle and also will determine the location and size of the obstacle based on the laser range finder detection data. Using data of the current location of the mobile robot, the location of the obstacle and the final point, the mobile robot will decide whether to turn right or left in order to avoid the obstacle and to optimize its newly generated trajectory in term of the smooth turning. Then the mobile robot will follow the newly generated trajectory until it reaches the final point.

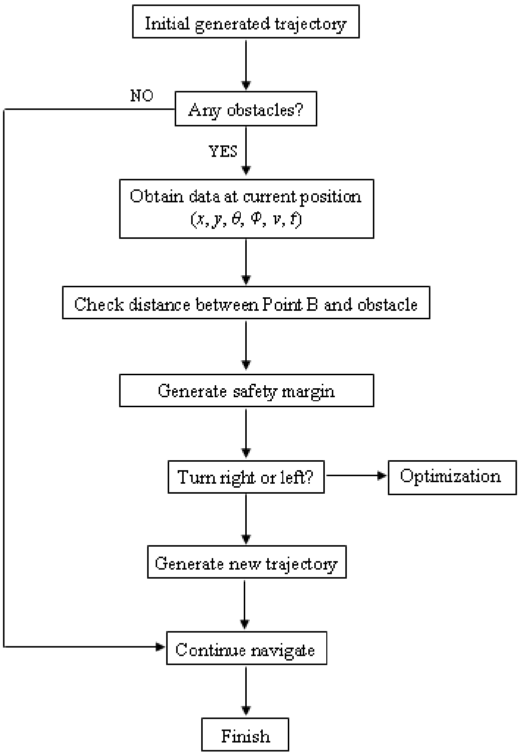


Figure 3 Algorithm steps flowchart for avoiding an obstacle.

In the case where the mobile robot has not detected any osbatcle, it will then continue to navigate along the initial trajectory until it reaches the final point.

Figure 4 illustrates how a mobile robot avoids a static obstacle. The illustration is based on the flowchart in Figure 3. Basically, safety allowance is the size of the mobile robot. The radius of safety margin is determined by:

$$rad_{sm} = rad_{obs} + \delta \qquad (15)$$

where,

rad_{obs} = radius of obstacle
δ = safety allowance

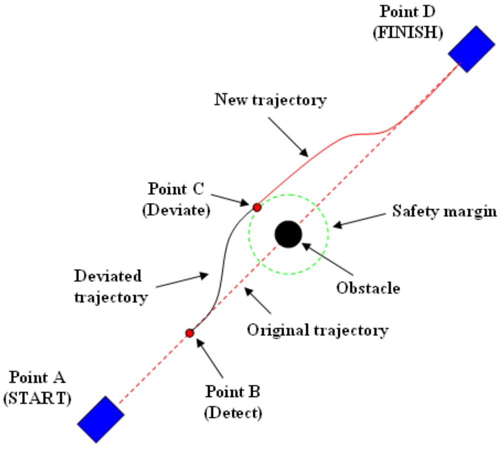


Figure 4 Mobile robot avoiding an obstacle [Hashim and Lu, 2009].

4. Simulation set-up, results and discussions

All the simulations were conducted in the Matlab software. The map was developed in graphic editor, using Microsoft Paint in this study. Several general assumptions were made for the modelled car-like robot in this simulation:

- The robot moves on horizontal plane.
- Single point contact of the wheel.
- The wheel is not deformable.
- No slipping, skidding or friction
- The wheels are attached at the rigid chassis
- Radius of the obstacle is known
- Shape of the obstacle is treated as symmetric and consistent

The physical data for the mobile robot are as follows:

- Length, $l = 1.3$ m
- Width, $w = 0.6$ m
- Rear wheel radius, ρ 0.3
- Maximum steering angle, ϕ = 20 degrees.

For the detection sensor, the laser range finder is simulated and the detection length and angle are 20 meters and 180 degrees, respectively, with angular resolution of 0.25°. For simulation purpose, the map size is set to be 100m x 100m and 60 seconds is assigned to reach the final point. The output (actual) data is collected at every one second in the simulation environment. Generally, in this simulation, the mobile robot is represented by a blue box with length of l and width of two times of w. The unknown static obstacles are represented by a 4m x 4m green rectangle.

The environment was setup as shown in Figure 5. The black areas represent the walls and/or known obstacles. The waypoints are represented by red circles. Point 1 and Point 4 are the initial and final points, respectively. Point 2 and Point 3 are the desired waypoints. The orientation at each points is indicated by an arrow. The input data for this simulation are summarized in Table 1. The control inputs for this simulation are steering angle and velocity.

Table 1: Input data for simulation

Point	t (sec)	x (m)	y (m)	θ (°)	ϕ (°)	v (m/s)
1	0	15	10	90	0	0
2	20	30	50	0	0	2
3	40	70	50	0	0	2
4	60	85	90	90	0	0

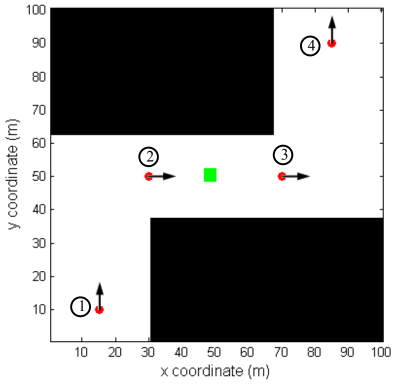
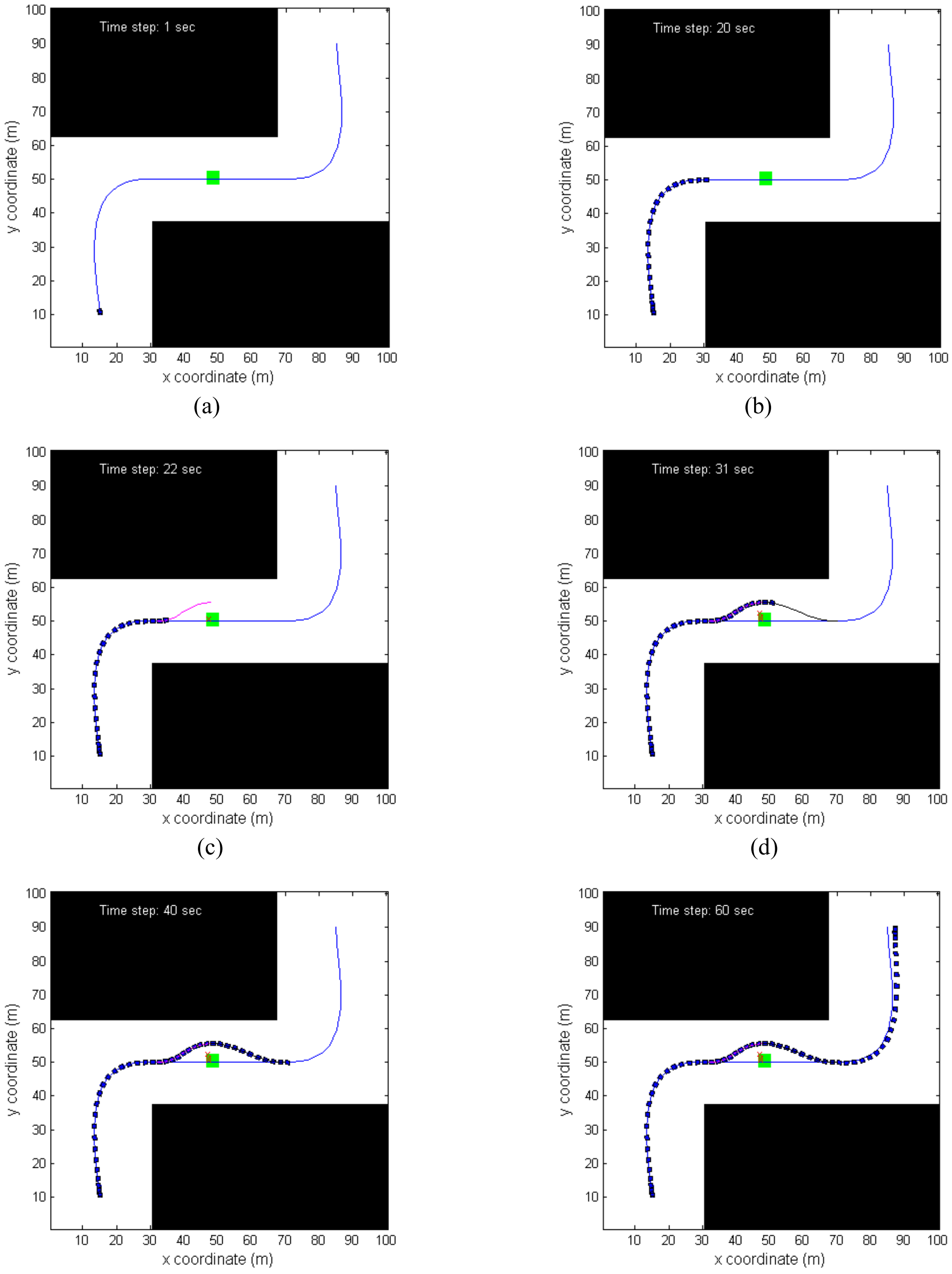


Figure 5 Prior map with two waypoints connecting the initial and final point.

The simulation algorithm consists of both offline and online planning components. The offline planning deals with the known obstacles, where it will be executed at the inital stage. Then, the online planning will be executed once the mobile robot starts navigating in the environment. The online planning is to detect and deal with unknown obstacles.

The simulation results are shown in Figure 6. At the initial stage, the offline planning is executed and the initial trajectory is generated, which is represented by a blue line as shown in Figure 6(a). Once the initial trajectory is generated, the mobile robot starts to move away from the initial point. Then the mobile robot moves along the initial trajectory until it reaches the first waypoint as shown in Figure 6(b). It then continues its journey until it detects obstacle, as in Figure 6(c). Once the obstacle is detected, the obstacle avoidance algorithm is executed. Using the actual robot data at the detection point, the deviation point will be determined. Then the deviation trajectory is generated from the detection point to the deviation point, as shown in Figure 6(c). And the new trajectory is generated from the deviation point to the second waypoint with adjusted travelling velocity, as shown in Figure 6(d). Finally the mobile robot continues its journey until it reaches the second waypoint and then the final point.



(a) (b) (c) (d) (e) (f)

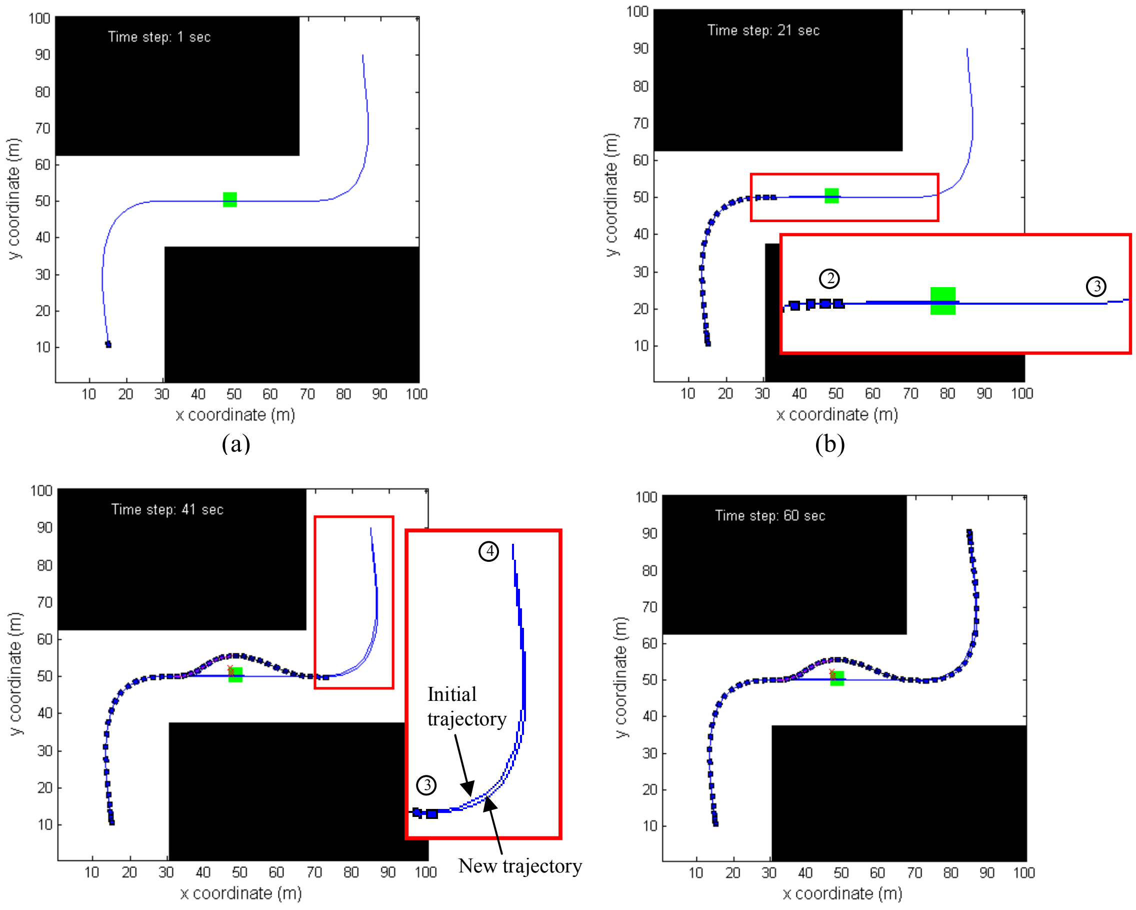
Figure 6 Simulation results (a) at time step = 1 sec, (b) at time step = 20 sec, (c) at time step = 22 sec, (d) at time step = 31 sec, (e) at time step = 40 sec and (f) at time step = 60 sec.

In conclusion, the multiple waypoints trajectory planning approach has been succesfully simulated for a car-like mobile robot in time-depended environment. The mobile robot was able to detect and avoid the obstacle smoothly. It was also able to catch-up the time delayed due to avoiding the obstacle before reaching the final point at specified time. Table 2 summarizes the actual collected data at every waypoint. In comparison to input data for simulation in Table 1, the errors in position and orientation at the final point are around 2.29 meters and 0.3 degrees, respectively. These errors are quite large, especially for position error.

Table 2 Actual collected data of simulation

Point	t (sec)	x (m)	y (m)	θ (°)	$\varnothing$ (°)	v (m/s)
1	0	15	30	90	0	0
2	20	30.08	49.99	2.1	0	2
3	40	70.00	49.97	-3.3	0	2
4	60	87.21	89.04	89.7	0	0

Therefore in order to reduce the errors at every waypoint, replanning approach is introduced. This approach uses the actual data, such as position, orientation, steering angle and velocity, at the current waypoint and replan the new trajectory to the next waypoint. This will ensure the errors are reduced. The simulation results with replanning approach are shown in Figure 7.



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

Figure 7 Simulation results with replanning approach (a) at time step = 1 sec, (b) at time step = 21 sec, (c) at time step = 41 sec and (d) at time step = 60 sec.

Figure 7(a) shows the initial planned trajectory from the initial point to the final point, and pass through all the waypoints. Then the mobile robot navigates along the initial planned trajectory until it reaches Point 2. Once it reaches Point 2, the replanning algorithm is executed. Using the actual data at Point 2, a new trajectory is generated from Point 2 to Point 3, as shown in Figure 7(b). The new trajectory is almost identical to the initial trajectory because the errors are quite small. Then the mobile robot continues its journey along the new trajectory, avoids the obstacle and reaches Point 3. At this point, replanning algorithm once again is executed and a new trajectory is generated from Point 3 to the final point, as shown in Figure 7(c). The mobile robot then continues its journey and finally reaches the final point, as shown in Figure 7(d). The actual collected data at every waypoint are summarized in Table 3.

Table 3 Actual collected data with replanning approach

Point	t (sec)	x (m)	y (m)	θ (°)	$\varnothing$ (°)	v (m/s)
1	0	15	30	90	0	0
2	20	30.08	49.99	2.1	0	2
3	40	70.00	49.97	-3.3	0	2
4	60	84.9	90.00	93.2	0	0

From the, the data at both waypoints - Point 2 and Point 3 - are not much different from data in Table 2. This is because at Point 2, the replanning approach was not yet been executed. While at Point 3, the trajectory was effected by the obstacle avoidance algorithm. Therefore the significant errors different can be perceived at the final point. With replannning approach, the errors in position and orientation are around 0.1 meters and 3.2 degrees, repectively. Although the error in orientation is greater than the previous orientation error,it is considered as still in satisfactory limits, with the consideration of both position and orientation errors.

5. Conclusion and future work

Besides a time-dependent motion planning, this paper has extended the initial work [Hashim and Lu, 2009] on trajectory generation using geometric approach by introducing a multiple waypoints trajectory planning approach which is able to specify desired specific position, orientation, velocity and arrival time of the mobile robot at every waypoint. Furthermore, replanning approach is introduced at every waypoint in order to reduce the errors in position and orientation. The simulation results demonstrate the practicality and effectiveness of this approach to be used in motion planning, especially for a car-like mobile robot. The authors are currently extending this work to deal with dynamic obstacles and multiple robots applications and taking the dynamic constraints of the mobile robot into account. Furthermore, experiments will be conducted to validate the algorithms.

References

[Dong and Guo, 2005] Wenjie Dong and Yi Guo, “New trajectory generation methods for nonholonomic mobile robots”, The International Symposium on Collaborative Technologies and Systems, Missouri, USA, May 2005.

[Guo *et al*., 2003] Yi Guo, Zhihua Qu and Jing Wang, “A new performance-based motion planner for nonholonomic mobile robots”, The 3rd Performance Metric for Intelligent System Workshop, Gaithersburg, USA, September 2003.

[Guo *et al*., 2007] Yi Guo, Yi Long and Weihua Sheng, “Global trajectory generation for nonholonomic robots in dynamic environments”, The 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, April 2007.

[Hashim and Lu, 2009] M. Sani M. Hashim and Tien-Fu Lu, “Time-dependent motion planning for nonholonomic mobile robots”, The 9th International IFAC Symposium on Robot Control (SYROCO), Gifu, Japan, September 2009.

[Jolly *et al*., 2009] 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.

[Kanayama and Miyake, 1986] Y. Kanayama and N. Miyake, “Trajectory generation for mobile robots”, The 3rd Symposium on Robotics Research, Gourvieux, France, 1986.

[Kelly, 2003] Alonzo Kelly, “Efficient parametric synthesis of optimal mobile robot trajectories”, The 11th International Conference on Advanced Robotics, University of Coimbra, Portugal, July 2003.

[Liddy and Lu, 2007] Tommie Liddy and Tien-Fu Lu, “Waypoint with position and heading control using complex vector fields for Ackermann steering autonomous vehicle”, The 2007 Australasian Conference on Robotics and Automation (ACRA), Brisbane, Australia, December 2007.

[Nagy and Kelly, 2001] Bryan Nagy and Alonzo Kelly, “Trajectory generation for car-like robots using cubic curvature polynomials”, The 3rd International Conference on Field and Service Robots, Helsinki, Finland, June 2001.