A New Performance-Based Motion Planner for Nonholonomic Mobile Robots

Yi Guo, Zhihua Qu and Jing Wang
School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL 32816-2450
Emails: yguo@ee.ucf.edu, qu@mail.ucf.edu, jwang@pegasus.cc.ucf.edu

Abstract

We address the global trajectory planning problem of nonholonomic mobile robots in environments with static and dynamic obstacles. The global trajectory is composed of regional feasible trajectories which satisfy the dynamics of the robot kinematic model. Piecewise constant parameterization is used to construct regional feasible trajectory and steering control, and collision avoidance criterion is derived. Performance considered in the paper include robot safety, geometry-based criteria, time-based criteria, and physics-based criteria. Regional analytic trajectory solutions facilitate performance evaluation of the global trajectory. Simulations show a good performance of the planned trajectory using the proposed scheme.

Keywords: *Motion planning, nonholonomic system, dynamic obstacle, parameterized trajectory, steering control.*

1 Introduction

In many applications of mobile robots such as surface mining and space exploration, a prior map exists at certain degree of accuracy. The performance in concern for the motion planner usually includes:

1. Safety: The robot should avoid any dangerous terrain which causes static instability of the robot by certain margins;
2. Geometry-based criteria: criteria that relate to the geometry, such as shortest distance;
3. Time-based criteria: criteria that are a function of time, such as shortest time;
4. Physics-based criteria: criteria depending on the physical configuration of the robot, such as optimal fuel or energy.

Although path planning and motion control are closely related in the robot navigation problem, they are usually treated as two separate problems in much of the existing literature. Path planning is the determination of the geometric path points for the mobile robots to track, and motion control is the determination of the physical input to the robot motion components. These issues are typically discussed using methods in different areas such as those in artificial intelligence and control theory. Such a separation makes it difficult to address robot performance in a complete application, since the discrete geometric path points planned in the first step cannot be efficiently tracked in the second step due to the nonholonomic motion constrains of the robots, thus losing the meaning of optimization in each step. In this paper, we propose a new scheme to generate global feasible trajectory for nonholonomic mobile robots, and provide preliminary evaluation of performance as defined above.

It is well known that motion planning for mobile robots has not been an easy task even in the absence of obstacles, due to the facts that nonholonomic constraints of mobile robots (kinematic constraints) make time derivatives of configuration variables of the system non-integrable, and any given path in the configuration space does not necessarily correspond to a feasible path for the nonholonomic system. To make the planned path trackable by nonholonomic robots, we introduce feasible trajectories which accounts for the kinematic constrains of the robots and are parameterized for a collision-free solution in the environment with both static and dynamic obstacles. And it is natural to expect that a global planned path is composed of segments of such feasible trajectories.

D* search ([1]) has been recognized as an effective global path searching method which returns sequences of path points in known or partially known environment. As used in [2], a complete navigation system should integrate the local and global navigation systems: the global system pre-plan a global path and incrementally search new paths when discrepancy with the map occurs; the local system uses onboard sensors to detect and avoid unpredictable obstacles. However, the steering arbiter method used in [2] simply votes a weighted sum of global and local steering commands, which is feasible in practice but provide no performance guarantees. A novel regional trajectory generation algorithm was recently proposed in [3] for nonholonomic mobile robots in a dynamic environment. Closed-form analytic solutions are provided by employing the polynomial inputs. In this paper, we provide an interface of combining the global path search and regional trajectory generation in environments containing static and dynamic obstacles, and provide analytic performance deviation analysis from the lower bound of optimal solutions. The paper combines recent techniques presented in [3, 4, 5, 6].

The remainder of the paper is organized as follows. In Section 2, the model of a nonholonomic car-like mobile robot is first transformed into the chained-form, and the feasible trajectory is defined satisfying both boundary conditions imposed and dynamics of the robot kinematic model. The steering paradigm is explicitly constructed for a given feasible trajectory. Then in Section 3, motion planning scheme will be presented in detail. Performance evaluation of the proposed motion planning scheme will be discussed in Section 4. In Section 5, simulation results will be shown. And finally we conclude with brief remarks in Section 6.

2 Problem Formulation and Parameterized Feasible Trajectories

Assumptions of our study are stated as follows:

Assumptions

1. The robot under consideration is represented by a 2-dimensional circle with radius R, as shown in Figure 1.

2. The robot operates in a 2D environment with static and dynamic obstacles. The $i$the obstacle is represented by a circle with radius r_i. For a moving obstacle, the center is time varying and moving with linear velocity vector.

3. A pre-defined map exists with static obstacle locations stored.

4. The robot has an assigned goal, and knows its start and goal positions.

5. The robot onboard sensors detect the dynamic obstacles as it goes.

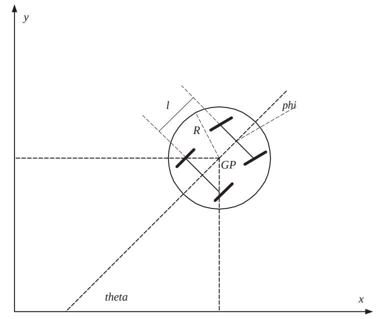

Figure 1: A car-like robot.

The problem under concern is defined as follows:

Find feasible trajectories for the robot, enrouting from its start position to its goal position, without collisions with static and dynamic obstacles, while satisfying both boundary conditions imposed and dynamics of the kinematic model of the nonholonomic motion.

2.1 Nonholonomic Kinematic Model

We consider a car-like mobile robot whose front wheels are steering wheels and rear wheels are driving wheels but have a fixed forward orientation. The state space representation of the kinematic model taking nonholonomic constrains is given by (see [3] for detail derivation):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{l}{2}\sin(\theta) & 0 \\ 0 & 1 & \frac{l}{2}\cos(\theta) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \rho\cos(\theta) & 0 \\ \rho\sin(\theta) & 0 \\ \frac{\rho}{l}\tan\phi & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{1}$$

where $q = [x\ y\ \theta\ \phi]$ is the state vector: $[x\ y]$ represents the Cartesian coordinates of the guide point, θ is the orientation of the robot body with respect to the x-axis, ϕ is the steering angle; l is the distance between the two wheel-axle centers, ρ is the radius of back driving wheel; u_1 is the angular velocity of the driving wheel, and u_2 is the steering rate of front guiding wheel. The range of ϕ is limited to be within $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ due to singularity, and θ is within $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ to ensure an one-to-one map of the coordinate transformation.

The kinematic model (1) can be transformed into the well-defined chained-form as:

$$\begin{aligned}\dot{z}_1 &= v_1\\ \dot{z}_2 &= v_2\\ \dot{z}_3 &= z_2 v_1\\ \dot{z}_4 &= z_3 v_1 \end{aligned} \qquad (2)$$

under the following coordinate and input transformations:

$$\begin{aligned} z_1 &= x - \frac{l}{2}\cos(\theta)\\ z_2 &= \frac{\tan(\phi)}{l\cos^3(\theta)}\\ z_3 &= \tan(\theta)\\ z_4 &= y - \frac{l}{2}\sin(\theta), \end{aligned} \qquad (3)$$

$$\begin{aligned} u_1 &= \frac{v_1}{\rho\cos(\theta)}\\ u_2 &= -\frac{3\sin(\theta)}{l\cos^2(\theta)}\sin^2(\phi)v_1\\ &\quad + l\cos^3(\theta)\cos^2(\phi)v_2. \end{aligned} \qquad (4)$$

As pointed out in [3], the chained forms such as (2) have been used as a canonical form to describe nonholonomic systems (see [7]), and different steering schemes can be applied for such chained form systems, such as, sinusoidal steering, polynomial steering, piecewise constant steering (see [8]). In the following, we use piecewise constant parameterization to define trajectory and steering control.

2.2 Parameterized Feasible Trajectories

As defined in [3], we say *a trajectory is feasible if it satisfies both boundary conditions imposed and dynamics of the kinematic model.*

Given a set of boundary conditions with

$$\begin{aligned} q(t_0) &= q^0 = [x_0, y_0, \theta_0, \phi_0],\\ q(t_f) &= q^f = [x_f, y_f, \theta_f, \phi_f], \end{aligned} \qquad (5)$$

the feasible trajectories are represented by

$$y = F(x - 0.5l\cos(\theta)) + 0.5l\sin(\theta); \qquad (6)$$

By (3), in the transformed z state space, the boundary conditions are

$$z^0 = [z_1^0, z_2^0, z_3^0, z_4^0], \qquad z^f = [z_1^f, z_2^f, z_3^f, z_4^f], \qquad (7)$$

and the feasible trajectories are

$$z_4 = F(z_1), \qquad (8)$$

where F is a parameterized sixth-order polynomial taking the following form

$$\begin{aligned} F(z_1) &= a f(z_1)\\ &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}\\ &\quad \cdot \begin{bmatrix} 1 & z_1 & z_1^2 & z_1^3 & z_1^4 & z_1^5 & z_1^6 \end{bmatrix}^T \end{aligned} \qquad (9)$$

By imposing the boundary points and their slopes and curvature (see [3] for detail derivations), the feasible trajectories are parameterized in terms of a_6 as

$$z_4 = \begin{bmatrix} B^{-1}(Y - Aa_6) \\ a_6 \end{bmatrix}^T f(z_1) \qquad (10)$$

where

$$A = \begin{bmatrix} (z_1^0)^6 \\ 6(z_1^0)^5 \\ 30(z_1^0)^4 \\ (z_1^f)^6 \\ 6(z_1^f)^5 \\ 30(z_1^f)^4 \end{bmatrix}, \quad Y = \begin{bmatrix} z_4^0 \\ z_3^0 \\ z_2^0 \\ y_f - l\sin(\theta_f)/2 \\ \tan(\theta_f) \\ \tan(\phi_f)/(l(\cos(\theta_f)^3)) \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & z_1^0 & (z_1^0)^2 & (z_1^0)^3 & (z_1^0)^4 & (z_1^0)^5 \\ 0 & 1 & 2(z_1^0) & 3(z_1^0)^2 & 4(z_1^0)^3 & 5(z_1^0)^4 \\ 0 & 0 & 2 & 6(z_1^0) & 12(z_1^0)^2 & 20(z_1^0)^3 \\ 1 & (z_1^f) & (z_1^f)^2 & (z_1^f)^3 & (z_1^f)^4 & (z_1^f)^5 \\ 0 & 1 & 2(z_1^f) & 3(z_1^f)^2 & 4(z_1^f)^3 & 5(z_1^f)^4 \\ 0 & 0 & 2 & 6(z_1^f) & 12(z_1^f)^2 & 20(z_1^f)^3 \end{bmatrix}.$$

It is shown in [3] that as long as $z_1^0 \neq z_1^f$, the feasible trajectories are achievable by steering paradigm (shown in the next subsection).

Note that in (10), a_6 is a free-chosen parameter which will be determined by collision avoidance criterion. By representing the feasible trajectories in terms of the parameter a_6 as (10), we increase the freedom of maneuver of the robot to account for the geometric constrains due to obstacles (which include static and dynamic obstacles).

2.3 Steering Paradigm

We apply a polynomial input for the steering paradigm, that is,

$$\begin{aligned} v_1(t) &= C,\\ v_2(t) &= C_0 + C_1 t + C_2 t^2 + C_3 t^3 \end{aligned} \qquad (11)$$

where C, C_0, C_1, C_2, C_3 are constants.

If we choose

$$v_1(t) = \frac{z_1^f - z_1^0}{T} \tag{12}$$

where T is the time that takes the robot to get to q^f from q^0. Then we can obtain the steering input $v_2(t)$ to achieve the feasible path (10) as the following

$$\begin{aligned} v_2(t) &= 6(a_3 + 4a_4 z_1 + 10a_5 z_1^2 + 20a_6 z_1^3)v_1 \\ &\quad +24(a_4 + 5a_5 z_1 + 15a_6 z_1^2)tv_1^2 \\ &\quad +60(a_5 + 6a_6 z_1)t^2 v_1^3 + 120a_6 t^3 v_1^4. \end{aligned} \tag{13}$$

The detail derivation to get (13) is given in [3].

Note that for any given feasible trajectories, the analytic steering function is explicitly constructed as above.

2.4 Obstacle Avoidance Criterion

2.4.1 Static Obstacles

As noted in Subsection 2.2, the open-choosing parameter a_6 provides a novel mechanism for combining the geometric constrains due to obstacles with the feasible trajectories. Given k static obstacles which is centered at $(x^i_{obstacle}, y^i_{obstacle})$, $i = 1, 2, \ldots, k$ in the environment, the collision avoidance criterion can be expressed as

$$\begin{aligned} &\min_{x\in[x_0,x_f]} (x - x^i_{obstacle})^2 + (y - y^i_{obstacle})^2 \\ &\geq (R + r_i)^2, \quad i = 1, 2, \ldots, k \end{aligned} \tag{14}$$

where (x, y) is the robot position along the feasible path, R is the radius of the robot, and r_i is the radius of the obstacles, assuming both the robot and obstacles as circles.

From the coordinate transformation (3), all possible locations of the point (x, y) are on the right semi-circle centered at (z_1, z_4) and of radius $l/2$ for $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. So the obstacle avoidance criterion (14) in the x-y plane can be transformed into the z plane as

$$\begin{aligned} &\min_{z_1\in[z_1^0,z_1^f]} (z_1 - x^i_{obstacle})^2 + (z_4 - y^i_{obstacle})^2 \\ &\geq (R + r_i + \frac{l}{2})^2, \quad i = 1, 2, \ldots, k \end{aligned} \tag{15}$$

Substitute the analytic express of the feasible trajectories (10) in to (15), we can obtain the following second-order inequality in terms of a_6:

$$\begin{aligned} &\min_{z_1\in[z_1^0,z_1^f]} g_2(z_1)a_6^2 + g_{1i}(z_1)a_6 + g_{0i}(z_1) \geq 0, \\ &i = 1, 2, \ldots, k \end{aligned} \tag{16}$$

where

$$\begin{aligned} g_2(z_1) &= \left[z_1^6 - \widetilde{f}(z_1)B^{-1}A\right]^2, \\ g_{1i}(z_1) &= 2\left[z_1^6 - \widetilde{f}(z_1)B^{-1}A\right] \\ &\quad \cdot \left[\widetilde{f}(z_1)B^{-1}Y - y^i_{obstacle}\right], \\ g_{0i}(z_1) &= \left[\widetilde{f}(z_1)B^{-1}Y - y^i_{obstacle}\right]^2 \\ &\quad +(z_1 - x^i_{obstacle})^2 - (R + r_i + \frac{l}{2})^2; \\ \widetilde{f}(z_1) &= \left[\begin{array}{cccccc} 1 & z_1 & z_1^2 & z_1^3 & z_1^4 & z_1^5 \end{array}\right]. \end{aligned} \tag{17}$$

Therefore by choosing the parameter a_6 to satisfy (16), the resulting trajectory is a feasible trajectory without collision to any of the static obstacles.

2.4.2 Dynamic Obstacles

To deal with the dynamic obstacles which is detected by the onboard sensors real time, a set of regional collision avoidance conditions can be derived based on equations (15) and (16). The idea is to express z_1 as a discrete function for each sampling period, and represent the obstacles by piecewise constant velocities. Then piecewise constant parameterization of the feasible trajectories can be obtained to get the closed-form solution. The detail derivation and results are given in [3].

3 Motion Planning Scheme

The motion planning scheme is divided into sequential modules which are described in the following subsections.

3.1 Global D* Search

The D^* search algorithm, which is a dynamic version of A^*, was proposed in [1, 9]. It produces an optimal path from the start position to the goal in the sense of minimizing a pre-defined cost function. It has the capability of rapid replanning, and has been used in real time planning in partially known environments with challenging terrains. As in A^*, its efficiency is highly determined by the chosen cost function. The cost function we use for D^* path planning is the following:

$$f_{pp} = \rho + d \tag{18}$$

where ρ is a large value if there is any obstacle penetrated by the path, and 0 otherwise; d is the Euclidean distance. Such a cost function guarantees that D^* returns an optimal path that avoids static obstacles, and is the shortest possible path if one exists. Note that the obstacle size is enlarged by the size of the robot and the safety margin pre-defined.

The output of this module is a path P, which consists of a sequence of geometric points (x, y), in the resolution of the grid of the environmental map.

Figure 2 shows such a path.

3.2 Way Points Generation

After we get the discrete path sequences, we want to generate a set of way points that will be good enough but not necessarily pass though every path points. An intuition of a "good" solution is that the way points are very close to the adjacent obstacle so that the constraints are active. Therefore, we choose k way points on the global path which are the closest points corresponding to each obstacle. If there are more than one points that are closest to the obstacle, we can pick the middle of them or any of them without sacrificing the performance much as long as the magnitude of obstacle sizes are relatively small in the global setting.

The output of this module is a set of points, and their slopes and curvatures:

$$(x_j, y_j, \theta_j, \phi_j), \quad j = 0, 1, 2, \ldots, k+1$$

with the first one is the global start point of the robot and the last one is the global goal point assuming there are k static obstacles along the path in the map.

3.3 Feasible Trajectories Generation

After the way points are available, we generate feasible trajectories taking every two adjacent way points as the boundary conditions using the techniques described in Sections 2.2 and 2.4. That is, for every segment of feasible trajectory, we need to choose the design parameter a_6 to ensure that the smooth trackable trajectory does not penetrate the static obstacles.

3.4 Avoiding Dynamic Obstacles Regionally

The onboard sensors of the robot detect the moving obstacles and the robot online re-generates the trajectory to avoiding the collision. This can be done by changing the design parameter a_6 accordingly which provides more freedom of maneuver for the robot. Assuming the robot sensor range is smaller than each pre-generated path segment denoted above, the techniques in [3] can be directly applied.

4 Performance Evaluation

4.1 Safety

Safety is quantified by *safety margin*. In the first module of D^* search, we enlarge each obstacle by a radius of $d_1 + R$, where d_1 is the so-defined "safety margin" and R is the robot radius, so that the returned path points and the way-points are at least $d_1 + R$ away from the obstacles. Then in the module of "feasible trajectory generation", we also include the safety margin d_1 in the r_i (the radius of the ith obstacle) term, so that from equation (14), the feasible trajectory is $d_1 + R$ away from the obstacles.

4.2 Path Length

The path length returned by D* search can be directly computed by

$$L_1 = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{19}$$

where (x_i, y_i) are the coordinates of path points returned by the D* search, and N is the number of points.

L_1 provides the lower bound of an optimal path (shortest path). If the grid resolution is fine enough comparing with the global magnitude, L_1 approximates to the shortest path. Note that in practice, to reduce the computational expense of the search, instead of using the regular grid, framed-quadtree can be used for the search, see [10].

Since the feasible trajectories generated are expressed by analytic polynomial functions, we can now calculate the length of the feasible trajectories.

For every feasible trajectory segment, the length of the trajectory is

$$L_2 = \int_{t_0}^{t_0+T} \sqrt{\dot{x}^2 + \dot{y}^2} dt \tag{20}$$

where (x, y) is on the feasible trajectory and satisfies the kinematic constrains of the nonholonomic systems. Due to the coordinate transformation described in Section 2, we can obtain

$$\begin{aligned} L_2 &\leq \int_{t_0}^{t_0+T} \left[\sqrt{\dot{z_1}^2 + \dot{z_4}^2} + \frac{l}{2} \left(\dot{\theta} \right) \right] dt \\ &= \int_{t_0}^{t_0+T} \sqrt{\dot{z_1}^2 + \dot{z_4}^2} dt + \frac{l}{2} \left(\theta_f - \theta_0 \right) \\ &= \int_{t_0}^{t_0+T} \dot{z_1} f(z_1) dt + \frac{l}{2} \left(\theta_f - \theta_0 \right) \end{aligned} \tag{21}$$

where

$$\begin{aligned} f(z_1) &= \Big[1 + \big(a_1 + 2a_2 z_1 + 3a_3 z_1^2 + 4a_4 z_1^3 \\ &\quad + 5a_5 z_1^4 + 6a_6 z_1^5 \big)^2 \Big]^{\frac{1}{2}} \end{aligned}$$

From (20) or (21), the deviation from the lower bound of the optimal solution L_1 can be calculated.

Since the way points generated by the first step D* search are based on shortest path criteria, the deviation of the length of the feasible trajectory is limited from the lower bound of shortest path. It should be pointed out that the feasible trajectories generated above using polynomials are not shortest paths between the two adjacent way points. Results on shortest path for nonholonomic robots are given in [11], where it is shown that shortest path motion could be achieved by concatenations of pieces from a 48 three-parameter trajectory sets. It is obviously not an easy task to generate such feasible shortest paths. We advocate the idea of finding sub-optimal solutions instead of an optimal one. The polynomial based trajectory presented above provides such a sub-optimal solution. Furthermore, it can avoid regional moving obstacles.

4.3 Time-Based Criteria

If the robot velocity is fixed, shortest distance path and shortest time path are equivalent. In the cases that quickest path is sought in terms of least zig-zag motion, different cost functions in D* search can be chosen so that the way points generated reflect such criteria. Due to space limit, such issues will be reported separately.

4.4 Physical-Based Criteria

Using the proposed scheme, steering controls u_1, u_2 in original coordinates can be obtained by submitting (12) and (13) into (4). Since steering controls are generated analytically, energy spent on the motion can be calculated based on the dynamic model of the robot (which is not presented here for the space limit). It is then possible to compare different trajectories based on energy efficiency.

In summary, the decomposition of the global path into regional segments and the analytic construction of regional trajectories facilities the performance analysis.

5 Simulations

In this section, we demonstrate simulation results for a nonholonomic mobile robot navigating in an environment with static and moving obstacles.

In Figure 2, it shows the path sequences by D* search, and the polynomial based feasible trajectory generated through way points. The global path is decomposed into five path segments, where the parameter a_6 is zero for each segment to account for the static obstacles. Note that since the range of definition for angles θ, ϕ is $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, the global coordinate needs to be rotated between the first three path segments, which causes trajectory discontinuity.

The path length returned by D* search is 63.80, and the length of the feasible trajectory is 61.86. It can be seen that the polynomial based trajectory is good enough to match the shortest path. (Note that the grid resolution for conducting D* search is not fine enough so the path points returned has bigger length.)

In Figure 3, moving obstacles are detected by robot onboard sensors while the robot navigates on the last two path segments. It shows that the original planned path collides with the moving obstacles. The circles are drawn every 5 seconds for the position of the robot (big circle) and the obstacle (small circle) with the same starting time.

The following setting are used in the simulation:

- Robot parameters: $R = 1, l = 0.8, \rho = 0.2$.
- Moving obstacles: $r_i = 0.5$, for $i = 1, 2$.
  Moving obstacle 1 (left): center: $[23, 15]$, velocity: $[0.1, 0.2]$.
  Moving obstacle 2 (right): center: $[45, 20]$, velocity: $[-0.1, -0.1]$.

In Figure 4, a new collision free path is shown. The calculated trajectory parameter a_6 for the second last segment is $9.4086*10^{-6}$, and for the last segment is $a_6 = 4.9973*10^{-6}$.

In Figure 5, the robot detects that obstacle 2 changes its velocity at $t_0 + 20$ seconds (t_0 is the time the robot starts on the same path segment), and re-calculate a new path accordingly (collision occurs if following the old path). The parameter settings used are:

- Center of obstacle 2: $[50, 20]$,
- Original velocity at $t = t_0$: $[-0.15, -0.1]$,
- New velocity at $t = t_0 + 20$: $[0.15, -0.29]$

The trajectory parameters a_6 for the last path segment are calculated to be:

- At $t = t_0$, $a_6 = 4.6054 * 10^{-6}$;
- At $t = t_0 + 20$, $a_6 = -5.3385 * 10^{-4}$.

The path length in this case is 75.03. We can see that by carefully choosing the design parameter a_6, dynamic obstacles are avoided.

Figures 6 and 7 show the time history of robot orientation angle $\theta(t)$, steering angle $\phi(t)$, and steering controls u_1, u_2 respectively, for the last two path segments according to the trajectory shown in Figure 5.

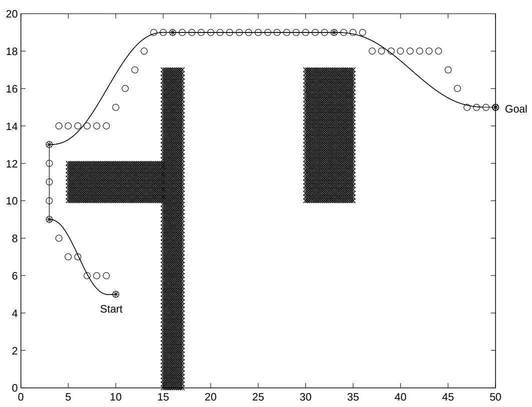

Figure 2: Global path: points denoted by circles are the path sequences returned by D* search; points denoted by solid circles are way-points; the solid line is global trajectory. The shaded areas are static obstacles.

6 Conclusions

In this paper, we have presented a new global motion planning algorithm for the nonholonomic mobile robots

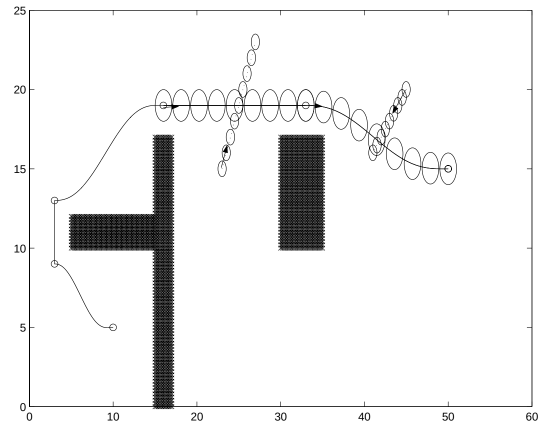

Figure 3: Collision path: motion of regional moving obstacles are denoted by series of small circles; motion of the robot is denoted by series of big circles. The arrow denotes the direction of motion.

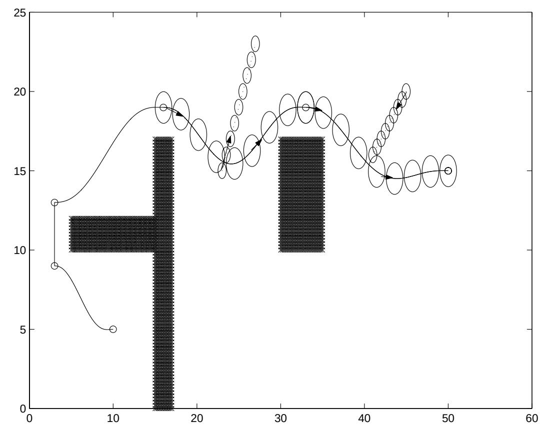

Figure 4: Collision-free path: constant speed moving obstacles.

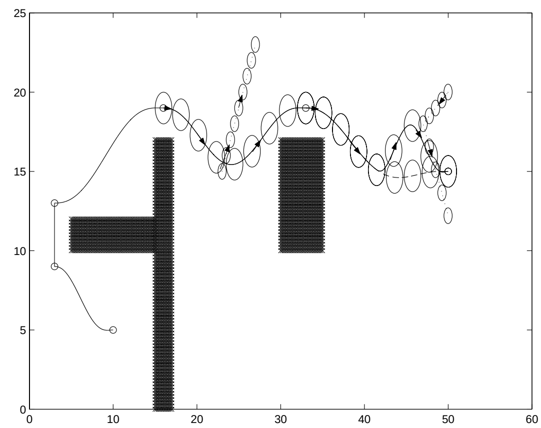

Figure 5: Collision-free path: obstacle changes velocity. Dashed line denotes old path; solid line denotes new path.

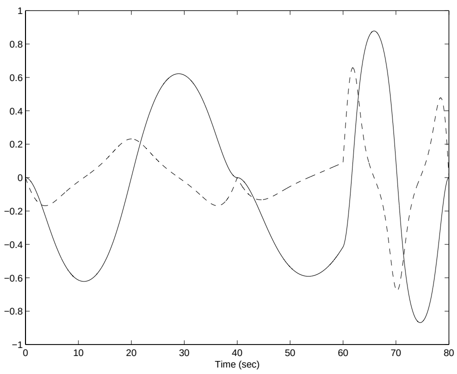

Figure 6: Robot orientation angle $\theta(t)$ (solid line) and steering angle $\phi(t)$ (dashed line) for the last two path segments.

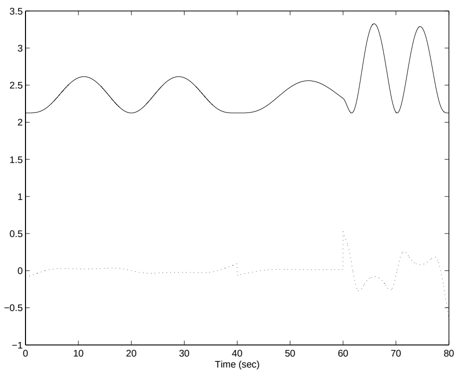

Figure 7: Steering controls: $u_1(t)$ (solid line) and $u_2(t)$ (dotted line).

in the presence of static and dynamic obstacles. Global D* search is firstly performed and discrete optimal path sequences are returned. Due to robot nonholonomic kinematics, the optimal path sequences can not be physically tracked by the robot. We then choose way-points from the global path and connect them using feasible trajectories. The feasible trajectory is defined to satisfy both boundary conditions imposed and dynamics of the robot kinematic model. Piecewise constant parameterization is used to construct feasible trajectory and steering control, and their solutions are obtained in closed-form. Representing moving obstacles by piecewise constant velocities, collision avoidance criterion is derived and combined in feasible trajectory construction. The regional analytic solutions facilitate performance evaluation of the global trajectory. The planned global trajectory is a sub-optimal solution, while the optimal one is very difficult to obtain for the nonholonomic mobile robots. Simulation results are shown and good performance is observed.

References

[1] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3310–3317, 1994.

[2] A. T. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, August 1995.

[3] Z. Qu, J. Wang, and C. E. Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. In *Proceedings of 2003 Florida Conference on Recent Advances in Robotics*, May 2003.

[4] Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2612–2619, 2002.

[5] L. E. Parker, K. Fregene, Y. Guo, and R. Madhavan. Distributed heterogeneous sensing for outdoor multi-robot localization, mapping, and path planning. In A. C. Schultz and L. E. Parker, editors, *Multi-Robot Systems: From Swarms to Intelligent Automata*, pages 21–30. Kluwer, 2002.

[6] L. E. Parker, Y. Guo, and D. Jung. Cooperative robot teams applied to the site preparation task. In *Proceedings of 10th International Conference on Advanced Robotics*, pages 71–77, 2001.

[7] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: steering using sinusoids. 38:700–716, 1993.

[8] D. Tilbury, R. M. Murray, and S. S. Sastry. Trajectory generation for the N-trailer problem using goursat normal form. 40(5):802–819, 1995.

[9] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995.

[10] A. Yahja, A. Stentz, S. Singh, and B. L. Brumitt. Framed-quadtree path planning for mobile robots operating in sparse environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, Belgium, May 1998.

[11] H. J Sussmann nad G. Wang. Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Technical Report SYSCON-91-10, Rutgers University, 1991.