2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Optimal Trajectory Generation for Nonholonomic Robots in Dynamic Environments

Yi Guo and Tang Tang

Abstract— **We study optimal trajectory generation for nonholonomic mobile robots in the presence of moving obstacles. The trajectory is presented by a parameterized higher-order polynomial and is feasible for car-like robots whose motion is nonholonomic. An optimal performance index is set up so that the parameterized trajectory stays close to the shortest straight-line path. Combining with the collision avoidance criterion, optimal collision-free trajectory can be generated real time as the solution is expressed in its closed-form. We show Matlab simulation results to demonstrate the performance of the trajectories.**

I. Introduction

Motion planning for car-like robots in dynamic environments is an inherently difficult problem. First, the paths have to be physically feasible and meet the nonholonomic constraint; and second, the generation of the path has to be real time to avoid moving obstacles. Trajectory generation for nonholonomic systems are discussed in many references including [1], [2], where trajectories are represented by sinusoidal, polynomial, or piecewise constant functions. In [3], quintic G_2 splines (splines with second-order geometric continuity) are constructed. Obstacle avoidance is not considered in the above work. On the other hand, many collision avoidance work (for example, [4], [5], [6], [7]) uses computational methods (such as A^* or D^*) whose real time performance may be constrained. More recently, analytic solutions of trajectory generation with moving obstacles are presented in [8], where parameterized polynomial trajectory is generated and collision avoidance criterion is developed based on it. The results of [8] meet the requirement of real time performance and apply to nonholonomic robots. Our paper is along the same line of the work [8] with a different sets of parameterized polynomial trajectories that are easier to incorporate into a global motion planning framework [9].

In our early paper [9], we presented a global trajectory generation method for nonholonomic robots in environments with moving obstacles. The method combines global path planning which generates global way-points with regional trajectory generation. Regional polynomial trajectories are feasible and represented in analytic solutions satisfying collision avoidance criterion. The polynomial trajectories presented are parameterized and have the flexibility for detour to avoid potential collisions. The current paper is a sequential work of [9], and we discuss the optimization problem for the parameterized polynomial trajectories. We minimize a performance index that is related to the area between our parametric trajectory and the shortest straight-line path, so that the long and unnecessary detour is avoided. We present optimal trajectory generation with and without obstacles. Collision avoidance criterion is incorporated so that moving obstacles can be avoided. Since closed-form solutions are developed, the trajectories can be generated real time without expensive computational cost. We show simulation results that verify the efficiency of the method.

II. SYSTEM MODEL AND PROBLEM STATEMENT

Consider a car-like mobile robot shown in Figure 1. The front wheels of the mobile robot are steering wheels and the rear wheels are driving wheels with a fixed forward orientation. The kinematic model of the mobile robot can be written as

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

where $q = [x, y, \theta, \phi]^T$ is the system state, (x, y) represents the Cartesian coordinates of the middle point of the rear wheel axle, 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 front and rear wheel-axle centers, u_1 is the driving velocity, and u_2 is the steering velocity. $\phi \in (-\pi/2, \pi/2)$ due to the structure constraint of the robot.

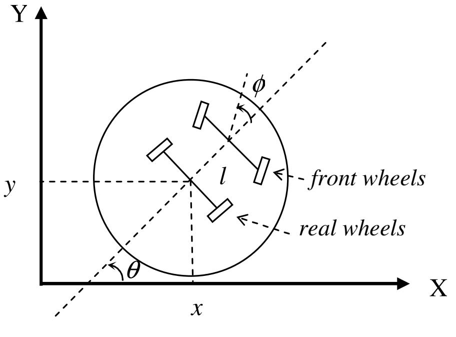

Fig. 1. A car-like robot.

We make the following assumptions on the robots and its environment.

Assumption 1: The robot is represented by a circle with the center at $O(t) = (x(t), y(t))$ and of radius R. Corresponding to the physical model shown in Figure 1, $O(t)$ is

Yi Guo and Tang Tang are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030. yguo1@stevens.edu, ttang1@stevens.edu

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

the reference point of the robot, and R is the radius of the minimum circle to bound the robot. The robot can locate itself.

Assumption 2: The robot has onboard sensors that detect the positions and velocities of obstacles. Specifically, at each sensor sampling time, kT_s $(k = 1, 2, \ldots)$, the robot detects the ith object centered at $O_i(t) = (x_i^o(t), y_i^o(t))$ and of radius r_i with velocity vector $v_i(t)$within its sensor range.

An illustration of the configuration of the robot and obstacles is shown in Figure 2.

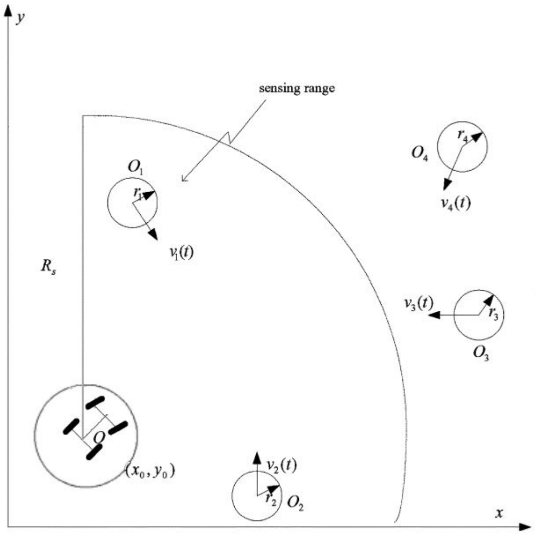

Fig. 2. Robot and obstacle configuration.

The problem we are going to solve is presented as below: *Given Assumptions 1 and 2, find a feasible optimal trajectory between two points $O_0 = (x_0, y_0)$ and $O_f = (x_f, y_f)$ from the initial time t_0 to the final time t_f, so that moving obstacles detected by the robot are avoided.*

In the above statement, "feasible trajectories" are defined to be smooth trajectories satisfying a given set of boundary conditions. We will discuss optimal trajectory generation (without obstacles) in the next section. Then, we will present optimal trajectory generation with obstacles in the following section.

III. OPTIMAL TRAJECTORY GENERATION

A. Parameterized Trajectory Generation

In [9], we proposed a parametric trajectory generation method that uses a sixth-order polynomial to represent the trajectory between two points $O_0 = (x_0, y_0)$ and $O_f = (x_f, y_f)$. Designating t_0, t_f as the initial and final time to get from O_0 to O_f, the trajectory is described as a function of t:

$$\begin{aligned} x(t) &= \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{bmatrix} f(t) \\ y(t) &= \begin{bmatrix} d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \end{bmatrix} f(t) \quad (2) \end{aligned}$$

where

$$f(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 & t^6 \end{bmatrix}^T, \quad (3)$$

and $c_0, c_1, \ldots, c_6, d_0, \ldots, d_6$ are constants.

For a set of given boundary conditions (with the initial time, t_0, and the final time, t_f, both given):

$$\left(x_0,\ y_0,\ \left.\frac{dx}{dt}\right|_{t_0},\ \left.\frac{dy}{dt}\right|_{t_0},\ \left.\frac{d^2x}{dt^2}\right|_{t_0},\ \left.\frac{d^2y}{dt^2}\right|_{t_0}, \right.$$
$$\left. x_f,\ y_f,\ \left.\frac{dx}{dt}\right|_{t_f},\ \left.\frac{dy}{dt}\right|_{t_f},\ \left.\frac{d^2x}{dt^2}\right|_{t_f},\ \left.\frac{d^2y}{dt^2}\right|_{t_f} \right), \quad (4)$$

if we represent the constant parameters $c_0, \ldots, c_5$ and $d_0, \ldots, d_5$ using c_6, d_6 respectively, we get the trajectory as a function of design parameters c_6, d_6:

$$\begin{aligned} x(t) &= \begin{bmatrix} G^{-1}(E - Hc_6) \\ c_6 \end{bmatrix}^T f(t) \\ y(t) &= \begin{bmatrix} G^{-1}(F - Hd_6) \\ d_6 \end{bmatrix}^T f(t) \quad (5) \end{aligned}$$

where

$$\begin{aligned} G &= \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \\ E &= \left[x_0\ x_f\ \left.\frac{dx}{dt}\right|_{t_0}\ \left.\frac{dx}{dt}\right|_{t_f}\ \left.\frac{d^2x}{dt^2}\right|_{t_0}\ \left.\frac{d^2x}{dt^2}\right|_{t_f} \right]^T \\ F &= \left[y_0\ y_f\ \left.\frac{dy}{dt}\right|_{t_0}\ \left.\frac{dy}{dt}\right|_{t_f}\ \left.\frac{d^2y}{dt^2}\right|_{t_0}\ \left.\frac{d^2y}{dt^2}\right|_{t_f} \right]^T \\ H &= \begin{bmatrix} t_0^6 & t_f^6 & 6t_0^5 & 6t_f^5 & 30t_0^4 & 30t_f^4 \end{bmatrix}^T \quad (6) \end{aligned}$$

The parametric trajectory (5) provides flexibility in designing optimal and collision-avoidance paths by changing the values of c_6 and d_6, and the steering control u_1 and u_2 are explicitly constructed in [9]. In the next subsection, we design optimal trajectory generation based on the above framework, and then in Section IV, we discuss obstacle-avoidance trajectories.

B. Optimal Trajectory Generation Without Obstacles

The strategy to avoid the long detour of paths is to obtain the shortest path between start point and end point [10]. The shortest path without constraints is the beeline between them, called 'initial straight line'.We can treat the straight line as reference line, and make the robot trajectories to be close to the reference line. For this purpose, we choose the performance index as :

$$J_k = \int_{t_0 + kT_s}^{t_f} \left| y(t) - \frac{y_f - y_0}{x_f - x_0}(x(t) - x_k) - y_k \right| dx \quad (7)$$

where T_s is the sensor sampling time and $k = 1, 2, \ldots$. The optimization problem is to make smallest the closure area enclosed by the feasible trajectories and the initial straight line, which makes sure the trajectory stays close to the initial straight line. Due to the difficulty in determining the sign of $[y - \frac{y_f-y_0}{x_f-x_0}(x(t)-x_k)-y_k]$, an optimal analytic solution based on (7) is difficult to obtain.

Motivated by the way of least square used in linear regression and variance analysis in statistics, we formulate the performance index as:

$$J_k = \int_{t_0+kT_s}^{t_f} [y(t) - \frac{y_f-y_0}{x_f-x_0}(x(t)-x_k)-y_k]^2 dx \tag{8}$$

The performance index is an integration to the square of the vertical coordinate difference between the class of parameterized trajectories and the initial straight line. Then the optimal problem is to minimize (8) subject to boundary conditions. This performance index can make the trajectory stay close to the initial straight line. In the following, we provide an analytical solution to the optimal trajectory.

Define

$$\bar{f}(t) = \left[\begin{array}{cccccc} 1 & t & t^2 & t^3 & t^4 & t^5 \end{array}\right]^T, \tag{9}$$

then

$$\bar{f}'(t) = [\ 0\ 1\ 2t\ 3t^2\ 4t^3\ 5t^4\ ]^T. \tag{10}$$

From (5), we get:

$$\begin{aligned} dx(t) &= \{(G^{-1}E)^T\bar{f}'(t) + [-(G^{-1}H)^T\bar{f}'(t) + 6t^5]\ c_6\} \\ & \quad dt \end{aligned} \tag{11}$$

Rewrite $x(t)$ and $y(t)$ as

$$\begin{aligned} x(t) &= h_1\bar{f}(t) + h_3c_6 \\ y(t) &= h_2\bar{f}(t) + h_3d_6 \end{aligned} \tag{12}$$

where

$$\begin{aligned} h_1 &= (G^{-1}E)^T \\ h_2 &= (G^{-1}F)^T \\ h_3 &= [-(G^{-1}H)^T\ 1]\left[\begin{array}{c} \bar{f}(t) \\ t^6 \end{array}\right] \end{aligned} \tag{13}$$

Let d_6 be a free-chosen variable, we have:

$$J_k = \int_{t_0+kT_s}^{t_f} [h_3d_6 + h_4]^2 h_5 dt \tag{14}$$

where

$$\begin{aligned} h_4 &= -vh_3c_6 + h_2\bar{f}(t) - vh_1\bar{f}(t) + vx_k - y_k \\ h_5 &= h_1\bar{f}'(t) + dh_3c_6 \\ v &= \frac{y_f-y_0}{x_f-x_0} \\ dh_3 &= [-(G^{-1}H)^T\ 1]\left[\begin{array}{c} \bar{f}'(t) \\ 6t^5 \end{array}\right] \end{aligned} \tag{15}$$

Let

$$\begin{aligned} f_1 &= \int_{t_0+kT_s}^{t_f} h_3^2h_5dt \\ f_2 &= 2\int_{t_0+kT_s}^{t_f} h_3h_4h_5dt \\ f_3 &= \int_{t_0+kT_s}^{t_f} h_4^2h_5dt \end{aligned} \tag{16}$$

If c_6 is known, J_k is a second order polynomial of d_6:

$$J_k = f_1d_6^2 + f_2d_6 + f_3, \tag{17}$$

whose minimal value is achieved when:

$$d_6 = -\frac{f_2}{2f_1} \tag{18}$$

Similarly, if d_6 is known and c_6 is free-chosen, we obtain

$$J_k = \int_{t_0+kT_s}^{t_f} [h_6c_6 + h_7]^2(h_8c_6 + h_9)dt \tag{19}$$

where

$$\begin{aligned} h_6 &= -vh_3 \\ h_7 &= h_3d_6 + h_2\bar{f}(t) - vh_1\bar{f}(t) + vx_k - y_k \\ h_8 &= dh_3 \\ h_9 &= h_1\bar{f}'(t) \end{aligned} \tag{20}$$

Let

$$\begin{aligned} I_1 &= \int_{t_0+kT_s}^{t_f} h_6^2h_8dt \\ I_2 &= 2\int_{t_0+kT_s}^{t_f} (h_6^2h_9 + 2h_6h_7h_8)dt \\ I_3 &= 2\int_{t_0+kT_s}^{t_f} (h_7^2h_8 + 2h_6h_7h_9)dt \\ I_4 &= \int_{t_0+kT_s}^{t_f} h_7^2h_9dt \end{aligned} \tag{21}$$

we obtain:

$$J_k = I_1c_6^3 + I_2c_6^2 + I_3c_6 + I_4 \tag{22}$$

After integration from t_0 to t_f, we get $I_1 = 0$. So if we know the value of d_6, we can also get a second order polynomial of c_6, whose minimal value is achieved when:

$$c_6 = -\frac{I_3}{2I_2} \tag{23}$$

IV. OPTIMAL COLLISION AVOIDANCE WITH MOVING OBSTACLES

In [9], we developed collision avoidance criterion for moving obstacles. To make the paper self-contained, we first present the collision avoidance criterion, and then discuss its optimization.

A. Collision Avoidance Criterion

Assume that the robot detects moving obstacles at time $t_0 + kT_s$ centered at $O_i^k = [x_i^k, y_i^k]$ with velocity $v_i^k = [v_{ix}^k, v_{iy}^k]$, where T_s is the sampling time of robot sensors. The collision avoidance criterion is to ensure that the future trajectory $x(t), y(t)$ for $t \in [t_0 + kT_s, t_f]$ does not collide with the obstacle. Note that the relative velocity between the robot and the obstacle is $[\dot{x} - v_{ix}^k, \dot{y} - v_{iy}^k]$. Taking the obstacle as "static", to avoid collision, whenever $x' \in [x_i^k - r_i - R, x_i^k + r_i + R]$, the distance between the centers of the robot and the obstacle must satisfy:

$$\left(x'(t) - x_i^k\right)^2 + \left(y'(t) - y_i^k\right)^2 \geq (r_i + R)^2 \tag{24}$$

where $x'(t) = x(t) - v_{ix}^k\tau, y'(t) = y(t) - v_{iy}^k\tau$ (relative position of the robot with respect to the "static" obstacle), $\tau = t - (t_0 + kT_s)$, and $t \in [t_0 + kT_s, t_f]$. Note that if $x' \notin [x_i^k - r_i - R, x_i^k + r_i + R]$, there won't be a collision.

An illustration of the collision avoidance scheme is shown in Figure 3.

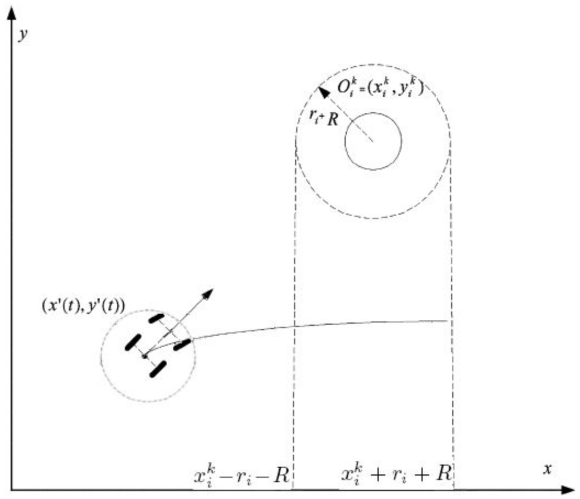

Fig. 3. Relative distance between the robot and the moving obstacle.

Substituting the trajectory expression (5) into (24), simplifying and reorganizing it, we obtain a second-order polynomial inequality in terms of c_6, d_6 as follows:

$$\begin{aligned}\min_{t\in[t_0+kT_s,t_f]} \quad & g_2(t)c_6^2 + g_1(t,\tau)c_6 + g_0(t,\tau) + h_2(t)d_6^2 \\ & +h_1(t,\tau)d_6 + h_0(t,\tau)|_{\tau=t-(t_0+kT_s)} \\ & -(r_i+R)^2 \geq 0 \qquad (25)\end{aligned}$$

where

$$\begin{aligned} g_2(t) &= (t^6 - \bar{f}G^{-1}H)^2 \\ g_1(t,\tau) &= 2(t^6 - \bar{f}G^{-1}H)(\bar{f}G^{-1}E - v_{ix}^k\tau - x_i^k) \\ g_0(t,\tau) &= (\bar{f}G^{-1}E - v_{ix}^k\tau - x_i^k)^2 \\ \bar{f} &= \left[\begin{array}{cccccc} 1 & t & t^2 & t^3 & t^4 & t^5 \end{array}\right], \\ h_2(t) &= g_2(t) = (t^6 - \bar{f}G^{-1}H)^2 \\ h_1(t,\tau) &= 2(t^6 - \bar{f}G^{-1}H)(\bar{f}G^{-1}F - v_{iy}^k\tau - y_i^k) \\ h_0(t,\tau) &= (\bar{f}G^{-1}F - v_{iy}^k\tau - y_i^k)^2 \qquad (26)\end{aligned}$$

We can now choose the design parameters c_6, d_6 to satisfy (25). Let $c_6 = 0$, since h_2 is positive, the following second-order polynomial in terms of d_6 has always solutions:

$$\begin{aligned}\min_{t\in[t_0+kT_s,t_f]} & h_2(t)d_6^2 + h_1(t,\tau)d_6 \\ & +[g_0(t,\tau) + h_0(t,\tau) - (r_i+R)^2]\big|_{\tau=t-(t_0+kT_s)} \geq 0.\end{aligned} \qquad (27)$$

Similarly, let $d_6 = 0$, we can get a set of values for c_6 to satisfy the collision avoidance criterion by solving

$$\begin{aligned}\min_{t\in[t_0+kT_s,t_f]} & g_2(t)c_6^2 + g_1(t,\tau)c_6 \\ & +[g_0(t,\tau) + h_0(t,\tau) - (r_i+R)^2]\big|_{\tau=t-(t_0+kT_s)} \geq 0.\end{aligned} \qquad (28)$$

Since (27) and (28) are second-order inequalities with coefficients of the second-order term positive, solutions always exist and are obtainable analytically. Note that the above process is iterated at every sampling time for $k = 1, 2, \ldots$ within $(t_0 + kT_s, t_f)$, so that all sensed obstacles are avoided.

B. Optimal Collision Avoidance

Based on the collision avoidance criterion presented above, we apply the optimal trajectory framework in Section III-B. We choose the values of c_6 or d_6 that are closest to the optimal values and also in the range for collision avoidance. That is, if the collision avoidance criterion generates a range of values for c_6 to be $\mathcal{A} = \{(-\infty, \underline{c}_6) \cup (\overline{c}_6, \infty)\}$, then we choose the $c_6 \in \mathcal{A}$ to yield $\min \|c_6 - c_6^{opt}\|$, where c_6^{opt} is the optimal c_6 generated using the optimal trajectory generation method presented in Section III-B. The reason for this choice is due to the parabolic function of c_6 in (22), which implies that closer c_6 is to c_6^{opt}, smaller J_k is.

The procedure to obtain the optimal trajectory with collision avoidance is summarized in the following:

- For the given boundary condition (4) and the initial time t_0 and the final time t_f, represent the trajectory by (5);
- Set $d_6 = 0$, use the optimal trajectory generation method of Section III-B (the equation (23)) to generate the optimal value c_6^{opt};
- Use the collision avoidance criterion (28) to find the range of values for c_6: $\mathcal{A} = \{(-\infty, \underline{c}_6) \cup (\overline{c}_6, \infty)\}$;
- Find the optimal c_6^* with collision avoidance to yield $\min \|c_6 - c_6^{opt}\|$;
- Set $c_6 = 0$, use the optimal trajectory generation method of Section III-B (the equation (18)) to generate the optimal value d_6^{opt};
- Use the collision avoidance criterion (27) to find the range of values for d_6: $\mathcal{B} = \{(-\infty, \underline{d}_6) \cup (\overline{d}_6, \infty)\}$;
- Find the optimal d_6^* with collision avoidance to yield $\min \|d_6 - d_6^{opt}\|$;
- Substitute c_6^* into (22) to get $J_k(c_6^*)$, and substitute d_6^* into (17) to get $J_k(d_6^*)$; if $J_k(c_6^*) < J_k(d_6^*)$, use $(c_6, d_6) = (c_6^*, 0)$ to compute the optimal collision-free trajectory (5); otherwise, use $(c_6, d_6) = (0, d_6^*)$.

Note that the optimal criterion we use is to minimize a performance index that is related to the area between the trajectory and the straight-line connecting the start and the goal points. The optimal trajectory may have more turns. Also, since we optimize one parameter while fixing the other, the method does not provide a global optimized solution, which is computational expensive and may not be practically applicable.

V. SIMULATION RESULTS

We have performed Matlab simulations to generate optimal collision-free trajectory. Figure 4 shows an optimal collision-free trajectory with two obstacles moving with constant velocities. The parameters we used are: $R = 0.3, r = 0.1, l = 0.2$, and the sensor range is 15. The starting positions of two obstacles O_1 and O_2 are $(2, 8.5), (5.5, 14)$, and their velocities are $V_x = 0.3, V_y = -0.4$. Both of Obstacle 1 and Obstacle 2 have potential collision with the nominal trajectory corresponding to $(c_6, d_6) = (0, 0)$ ($Path_1$ in the figure). Using the collision avoidance criterion, we get the range of values for c_6 is $(-\infty, -3.8346\times10^{-5})\cup(2.9074\times10^{-5}, \infty)$. Using $c_6 = 2.9074 \times 10^{-5}$, the collision-free path is shown as $Path_2$, the dashdotted line in the figure. Using the optimal trajectory generation method, we obtain $c_6^{opt} = -1.797 \times 10^{-5}$. The optimal trajectory is shown as $Path_3$, the solid line in the figure. But $Path_3$ collides with the obstacles. Following the procedure presented in Section IV-B, we choose $c_6^* = -3.8346\times10^{-5}$, which generates the optimal collision-free trajectory $Path_4$, the solid line with circles in the figure. We can see that $Path_4$ has short detour than the non-optimal collision-free path $Path_2$.

In Figure 5, we show the paths generated using two sets of parameters $(c_6, d_6^*) = (0, -2.946 \times 10^{-5})$ ($Path_1$) and $(c_6^*, d_6) = (-3.8346 \times 10^{-5}, 0)$ ($Path_2$). We can see that $Path_2$ has shorter detour than $Path_1$, which is confirmed by smaller J_k ($J_k(c_6^*) = 25.47, J_k(d_6^*) = 124.2$). It indicates that the path generated using c_6^* is optimal.

Note that the method applies to the cases where obstacles changes their velocities since the trajectory is expressed for each sensor sampling period in the optimal criterion (7) and the collision avoidance criterion (25). To illustrate it, Figure 6 shows an optimal collision avoidance path (the solid line $Path_1$) that avoids moving obstacles that changes their velocities. To make a comparison, $Path_2$ and $Path_3$ (the dashdotted lines in the figure) are collision-free trajectories without optimization. The parameters used are: when the robot is at $(16, 18.9)$, its sensor detects the obstacles at $O_1 = (2, 8.5)$ with velocity $(v_x, v_y) = (-0.2, 0.32)$, and $O_2 = (5.5, 14)$ with velocity $(v_x, v_y) = (-0.1, 0.2)$; after three sampling time, the robot detects that O_2 changes its velocity to $(v_x, v_y) = (0.55, 0.12)$. The trajectory parameter for $Path_2$ (the non-optimal collision-free path) is $(c_6, d_6) = (0, 0)$ and then $(0, 5.0895\times10^{-5})$ after detecting the changing velocity of O_2. The parameter for $Path_1$ (the optimal collision-free path) is $(c_6, d_6) = (0, -1.512 \times 10^{-6})$ and then $(0, 6.7359 \times 10^{-6})$. When the robot is at $(33, 19)$, its sensor detects the obstacles at $O_3 = (36, 17)$ with velocity $(v_x, v_y) = (0.6, -0.1)$, and $O_4 = (38, 18.25)$ with velocity $(v_x, v_y) = (0.1, 0.1)$; and then after three sampling time, the robot detects that O_3 changes its velocity to $(0.5, 0.08)$; after another two sampling time, the robot detects that O_4 changes its velocity to $(0.35, 0.15)$. The trajectory parameter for $Path_3$ (the non-optimal collision-free path) is $(c_6, d_6) = (0, 0)$ and $(0, -2.5966 \times 10^{-5})$ after detecting the changing velocity of O_3 and then $(0, -2.5966\times10^{-5})$ after detecting the changing velocity of O_4. The parameter for $Path_1$ (the optimal collision-free path) is $(c_6, d_6) = (0, -2.1077\times10^{-7})$ and $(5.0668 \times 10^{-5}, 0)$ after detecting the changing velocity of O_3 and then $(0, -2.6096 \times 10^{-5})$ after detecting the changing velocity of O_4. We can see that the optimal trajectory is more efficient.

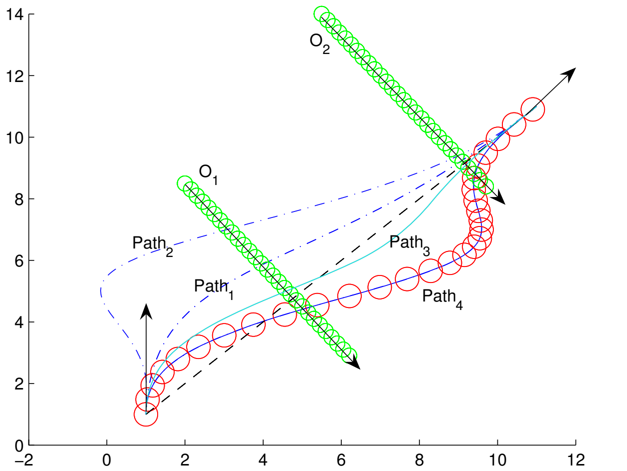

Fig. 4. Optimal collision-free trajectory $Path_4$ with comparisons to optimal path $Path_3$ and collision-free path $Path_2$. O_1 and O_2 denote obstacles with constant velocities. Circles are drawn with the same time interval.

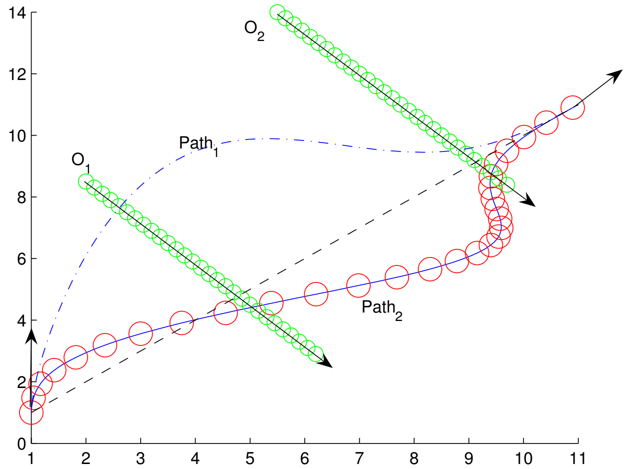

Fig. 5. Optimal collision-free trajectories: $Path_1$ is generated by using the optimal criterion on the parameter d_6, and $Path_2$ is by using the optimal criterion on the parameter c_6. Circles are drawn with the same time interval.

VI. CONCLUSIONS

We have studied optimal trajectory generation for car-like robots in the paper. The robot is assumed to have onboard sensors that detect the positions and velocities of moving obstacles within its sensor range. Optimal trajectory generation methods are presented for both obstacle-free and

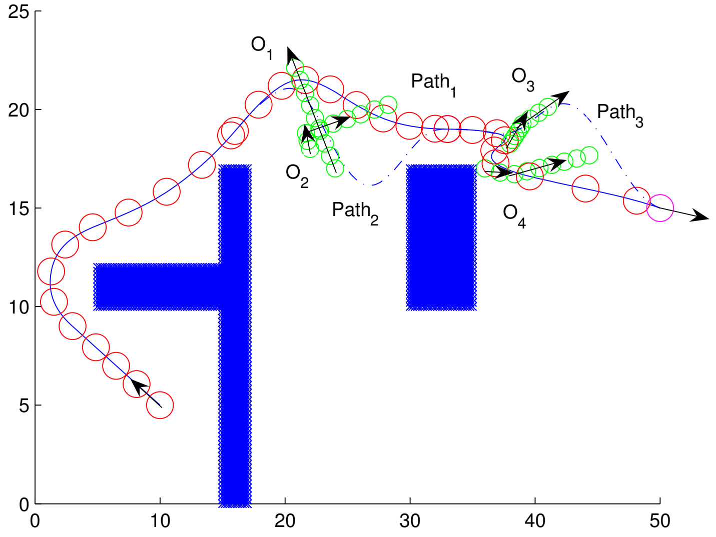

Fig. 6. Global optimal collision-free trajectories. Rectangular blocks denote stationary obstacles. Circles are drawn with the same time interval. $Path_1$ is the optimal collision-free path, and $Path_2$ and $Path_3$ are the non-optimal collision-free paths.

moving-obstacle cases. The optimal trajectory minimizes a performance index that is related to the area between the trajectory and the shortest straight-line path. Through the parameterized polynomial representation, closed-form solutions are found for optimal collision avoidance trajectories. Simulation results verify our claim.

References

[1] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: steering using sinusoids. *IEEE Trans. Automat. Contr.*, 38:700–716, 1993.

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

[3] A. Piazzi, C. G. L. Bianco, M. Bertozzi, A. Fascioli, and A. Broggi. Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 3(2):27–36, 2002.

[4] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1419–1424, 1986.

[5] K. Kant and S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.

[6] L. E. Parker. A robot navigation algorithm for moving obstacles. Master's thesis, The University of Tennessee, Knoxville, 1988.

[7] 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, May 2002.

[8] Z. Qu, J. Wang, and C. E. Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, 20(6):978–993, 2004.

[9] Y. Guo, Y. Long, and W. Sheng. Global trajectory generation for nonholonomic robots in dynamic environments. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1324–1329, April 2007.

[10] J. Yang, A. Daoui, Z. Qu, J. Wang, and R. A. Hull. An optimal and real-time solution to parameterized mobile robot trajectories in the presence of moving obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4412–4417, April 2005.