

Global Trajectory Generation for Nonholonomic Robots in Dynamic Environments

Yi Guo, Yi Long and Weihua Sheng

Abstract—We consider the problem of generating global feasible trajectories for nonholonomic mobile robots in the presence of moving obstacles. The global trajectory is composed of regional path segments, which are parametric polynomials incorporating collision avoidance criteria. Collision avoidance with moving obstacles is achieved by changing parameters of the regional trajectories, and the collision avoidance parameters are solutions to a second-order inequality and are obtainable analytically. To have a smooth global trajectory, we also develop a smooth irregular curve method to generate continuous boundary conditions to connecting regional trajectories. Steering control laws are constructed by means of differential flatness. The proposed technique works in dynamic environments where a set of global way-points are available, and the velocities of the obstacles are obtainable real time. Simulation results show the effectiveness of the methods.

Keywords: Motion planning, trajectory generation, collision avoidance, moving obstacles, steering control.

I. INTRODUCTION

In many applications of autonomous mobile robots, the robot needs to plan a global path and execute on it. Path planning is to determine a sequential set of discrete path points leading to the goal from the start position. There are mature path planning algorithms in an environment with stationary obstacles, such as road map, cell decomposition, heuristic search (A* and D* [14]), and potential field methods, see [6]. Motion planning in a dynamic environment is an inherently difficult problem. It involves prediction or calculation of future trajectories of moving obstacles. Motion planning methods reported in the presence of moving obstacles include discretization of configuration-time space [1], path and velocity planning decomposition [5], velocity obstacle [10], kinodynamic planning [7], and recently parameterized trajectory generation [12], to name a few.

Many autonomous robots have nonholonomic motion constraints, and any given path in a configuration space does not necessarily correspond to a physically feasible path for the robot. Path planning and motion control are usually treated in different subject areas such as artificial intelligence and control theory. Trajectory generation for nonholonomic systems is discussed in [8], [16], where steering control was designed and represented by sinusoidal, polynomial, or piecewise constant functions. The analytic trajectory generation methods works for environments without obstacles. As

studied in [13], [15], optimal path are very difficult to find for nonholonomic robots.

Continuous-curvature paths for autonomous vehicles was presented in [9], where path primitives were used on lane changing maneuvers and symmetric turns. In [11], global path planning is conducted by machine vision based methods, and quintic G^2 -splines (splines with second-order geometric continuity) are constructed for iterative steering of autonomous vehicles. Moving obstacles were not considered in the work.

In this paper, we study global motion planning strategies that combine benefits of heuristic and analytic methods. We argue that path planning methods such as A* and D* are efficient in a global scale at a coarse resolution to save computational expenses, and analytic methods are good at a regional scale for real time control. The idea was first presented in [3] by the first author and her co-authors. Global way-points was selected to be close to static obstacles so that constraints of static obstacles are active, and regional trajectory generation was applied between adjacent way-points. While it focuses on the general framework and discusses performance issues in [3], we present analytic solutions of trajectory generation at a global scale in this paper. In order to find the solution, we first solve a sub-problem, which is to generate feasible trajectories between two way-points given a set of boundary conditions in the presence of moving obstacles. Then, we generate continuous boundary conditions using a method for smooth irregular curves. Continuous steering control laws are constructed by means of differential flatness. Simulation results show the effectiveness of the proposed methods.

The rest of the paper is organized as follows. Section II presents the kinematic model of nonholonomic robots, and two problems are formulated towards generating global collision-free trajectories. In Section III, parameterized trajectories are presented, and steering control laws are constructed in Section IV. In Section V, collision avoidance criterion is presented together with solutions to Problem 1. A global smooth curve method is presented in Section VI and solutions to Problem 2 are presented. Simulation results are given in Section VII. The paper is finally concluded with brief remarks in Section VIII.

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

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

Weihua Sheng is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078.

be written as

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

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.

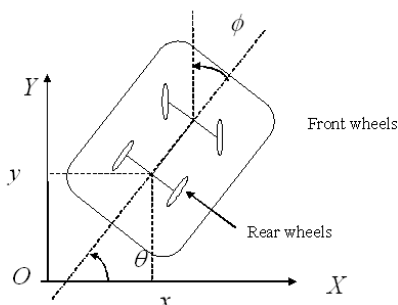


Fig. 1. A car-like robot.

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 the reference point of the robot, and R is the radius of the minimum circle to bound the robot.

Assumption 2: The robot has onboard sensors that detect the positions and velocities of obstacles. Specifically, at each sampling time of the sensors, the robot detects the i th 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.

Assumption 3: A set of way-points (x_i, y_i) , $i = 1, \dots, N$ are available by a global path planner.

We define the following two problems to be solved:

Problem 1: Given Assumptions 1 and 2, find a feasible trajectory between two points and the steering control input u_1, u_2 that avoids obstacles detected by the robot. Feasible trajectories are defined to be smooth trajectories satisfying a given set of boundary conditions. By smooth trajectory, we mean that the trajectory has continuous second-order derivatives at every point of the trajectory.

Problem 2: Given Assumptions 1 to 3, find a smooth collision-free trajectory and steering control u_1, u_2 to connect global way-points from (x_1, y_1) to (x_N, y_N) .

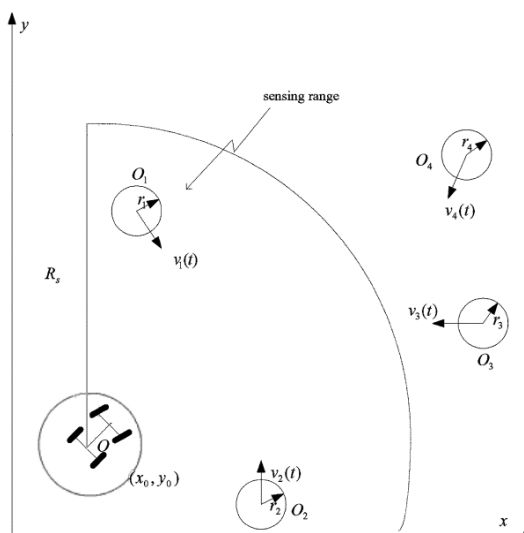


Fig. 2. Robot and obstacle configuration.

Problem 1 focuses on collision avoidance in the presence of moving obstacles. Problem 2 generates global smooth trajectories.

In the next, we consider Problem 1 first, and then solve for Problem 2 based on the result of Problem 1.

III. PARAMETERIZED TRAJECTORY GENERATION

We use a sixth-order parametric polynomial to represent the trajectory between two points $O_0 = (x_0, y_0)$ and $O_f = (x_f, y_f)$, and we designate t_0, t_f as the initial and final time to get from O_0 to O_f , respectively. The trajectory is described as a function of t :

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

where

$$f(t) = [1 \ t \ t^2 \ t^3 \ t^4 \ t^5 \ t^6]^T, \quad (3)$$

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

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

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

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

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

where

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

We will choose the parameters c_6, d_6 in Section V using the collision avoidance criterion.

IV. STEERING CONTROL DESIGN

The nonholonomic car-like robot (1) satisfies the property of differential flatness¹. From (1), we obtain:

$$\begin{aligned}
 \frac{dy}{dx} &= \tan \theta \\
 \frac{d^2y}{dx^2} &= \frac{\tan \phi}{l \cos^3 \theta} \quad (7)
 \end{aligned}$$

It is clear that θ, ϕ can be represented by the derivatives of the flat outputs, defined as x, y . Substituting the trajectory representation (2), we get:

$$\begin{aligned}
 \frac{dy}{dx} &= \frac{d_1 + 2d_2t + 3d_3t^2 + 4d_4t^3 + 5d_5t^4 + 6d_6t^5}{c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4 + 6c_6t^5}, \\
 \frac{d^2y}{dx^2} &= \frac{2d_2 + 6d_3t + 12d_4t^2 + 20d_5t^3 + 30d_6t^4}{2c_2 + 6c_3t + 12c_4t^2 + 20c_5t^3 + 30c_6t^4} \\
 \theta &= \arctan \frac{dy}{dx}, \\
 \cos \theta &= \sqrt{\frac{1}{1 + \left(\frac{dy}{dx}\right)^2}}, \\
 \phi &= \arctan \left(l \cos^3 \theta \cdot \frac{d^2y}{dx^2} \right), \\
 u_1 &= \frac{c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4 + 6c_6t^5}{\cos \theta}, \\
 u_2 &= \frac{1}{1 + \left(l \cos^3 \theta \cdot \frac{d^2y}{dx^2} \right)^2}. \quad (8)
 \end{aligned}$$

The analytic solution of the steering control u_1, u_2 drives the robot on the trajectory (2) from t_0 to t_f .

Note that θ has to be within the range of $(-\frac{\pi}{2}, \frac{\pi}{2})$ to avoid singularity of u_1 . In the case that the θ is out of the range, a rotation of the global axis can be performed and the process continues in the new global coordinate.

¹A system is defined differential flatness if there exists a set of outputs, such that all states and inputs can be expressed in terms of the outputs and their finite-order derivatives ([2]).

V. COLLISION AVOIDANCE CRITERION

To develop the collision avoidance criterion, we use similar arguments as in [12]. 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:

$$(x'(t) - x_i^k)^2 + (y'(t) - y_i^k)^2 \geq (r_i + R)^2 \quad (9)$$

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.

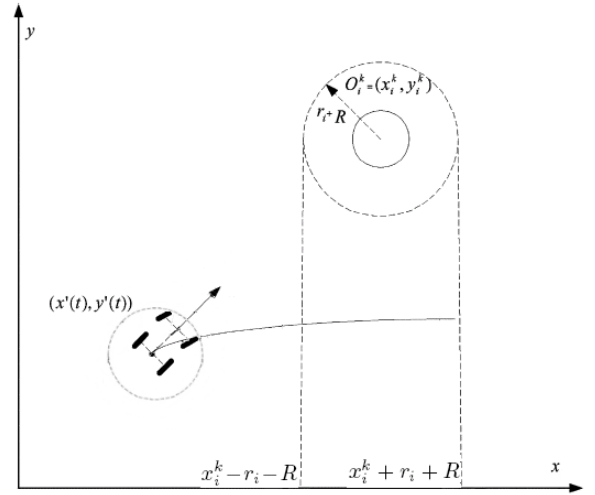


Fig. 3. Relative distance between the robot and the moving obstacle.

Substituting the trajectory expression (5) into (9), simplifying and reorganizing it, we obtain a second-order polynomial inequality in terms of c_6, d_6 as follows:

$$\begin{aligned}
 g_2(t)c_6^2 + g_1(t)c_6 + g_0(t) + h_2(t)d_6^2 + h_1(t)d_6 + h_0(t) \\
 - (r_i + R)^2 \geq 0 \quad (10)
 \end{aligned}$$

where

$$\begin{aligned}
 g_2(t) &= (t^6 - \bar{f}G^{-1}H)^2 \\
 g_1(t) &= 2(t^6 - \bar{f}G^{-1}H)(\bar{f}G^{-1}E - v_{ix}^k \tau - x_i^k) \\
 g_0(t) &= (\bar{f}G^{-1}E - v_{ix}^k \tau - x_i^k)^2 \\
 \bar{f} &= [1 \ t \ t^2 \ t^3 \ t^4 \ t^5], \\
 h_2(t) &= g_2(t) = (t^6 - \bar{f}G^{-1}H)^2 \\
 h_1(t) &= 2(t^6 - \bar{f}G^{-1}H)(\bar{f}G^{-1}F - v_{iy}^k \tau - y_i^k) \\
 h_0(t) &= (\bar{f}G^{-1}F - v_{iy}^k \tau - y_i^k)^2 \quad (11)
 \end{aligned}$$

We can now choose the design parameters c_6, d_6 to satisfy (10). Let $c_6 = 0$, since h_2 is positive, the following second-order polynomial in terms of d_6 has always solutions:

$$h_2 d_6^2 + h_1 d_6 + [g_0 + h_0 - (r_i + R)^2] \geq 0. \quad (12)$$

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

$$g_2 c_6^2 + g_1 c_6 + [g_0 + h_0 - (r_i + R)^2] \geq 0. \quad (13)$$

Since (12) and (13) 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 $k = 1, 2, \dots$ within $(t_0 + kT_s, t_f)$, so that all sensed obstacles are avoided.

We summarize the results of Sections III, IV, V in the following theorem:

Theorem 1: Problem 1 is solved by the parameterized trajectory (2) and the steering control (8), with c_6, d_6 satisfying the second-order polynomial inequality (10).

Remark 1: The above design provides options for collision-free trajectories. Let's denote the trajectory with $c_6 = d_6 = 0$ as the nominal trajectory corresponding to no moving obstacles detected. By choosing zero c_6 and non-zero d_6 , collision avoidance is achieved by altering the trajectory in the y direction. In contrast, choosing zero d_6 and nonzero c_6 , the trajectory in the x direction is altered to avoid potential collisions. In the simulation, we choose the values of c_6 or d_6 that are closest to their nominal values, 0. Further study on the performances (such as shortest path) needs to be done and will be in our future research.

Figures 4 and 5 show the collision-free paths with different choices of the design parameters c_6, d_6 . We can see that collision avoidance is achieved by changing the path in Figure 4, and by changing the velocity in Figure 5.

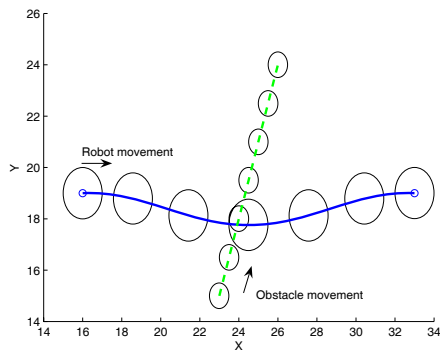


Fig. 4. Collision-free path (solid) and obstacle path (dashed). Circles are drawn in a 5-second interval. $c_6 = 0, d_6 = 1.0877 \times 10^{-7}$.

VI. GLOBAL SMOOTH CURVES

For a set of discrete way-points, we need to determine continuous boundary conditions in (4) so that the global trajectory is smooth at every point. Inspired by the method

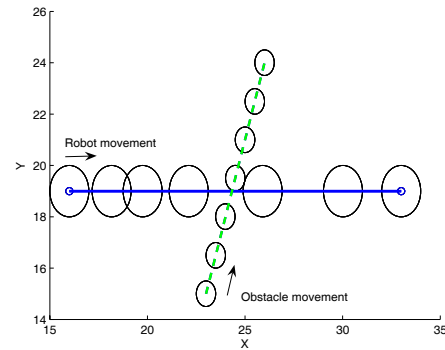


Fig. 5. Collision-free path (solid) and obstacle path (dashed). Circles are drawn in a 5-second interval. $d_6 = 0, c_6 = 2.0874 \times 10^{-7}$.

presented in [4] for generating smooth irregular curves, we propose the following procedure.

We consider a local subset of six points to define sequentially a local polynomial approximation to the curve between the third and fourth points in the local subset. After each step, a new point is added and the first point is deleted. It iteratively generates polynomial segments between adjacent points and there is no slope discontinuity.

We designate the time to reach each way-point from the beginning to be $0, t_2, t_3, t_4, \dots$, and use the time t as the parameter, which is single-valued and monotonically increasing. Use cubic polynomials to generate three curves: the left, which uses the first 4 points in the data subset, the middle, which uses the middle 4 points, and the right, which uses the last 4 points in the 6-point data subset. Since boundary conditions are well-defined as:

$$\begin{aligned} (t, x, y)_L &= \{0, x_1, y_1; t_2, x_2, y_2; t_3, x_3, y_3; t_4, x_4, y_4\}, \\ (t, x, y)_M &= \{t_2, x_2, y_2; t_3, x_3, y_3; t_4, x_4, y_4; t_5, x_5, y_5\}, \\ (t, x, y)_R &= \{t_3, x_3, y_3; t_4, x_4, y_4; t_5, x_5, y_5; t_6, x_6, y_6\}, \end{aligned}$$

the cubic polynomials are generated as

$$\begin{aligned} x &= a_1 + a_2 t + a_3 t^2 + a_4 t^3, \\ y &= b_1 + b_2 t + b_3 t^2 + b_4 t^3, \end{aligned} \quad (14)$$

where the constant coefficient matrices can be calculated as

$$\begin{aligned} A &= T^{-1}X \\ B &= T^{-1}Y, \end{aligned} \quad (15)$$

and T and X are matrices defined from the boundary conditions of each of the *Left, Middle, Right* polynomials:

$$\begin{aligned} A_L &= [a_{1L} \ a_{2L} \ a_{3L} \ a_{4L}]^T, \ B_L = [b_{1L} \ b_{2L} \ b_{3L} \ b_{4L}]^T; \\ A_M &= [a_{1M} \ a_{2M} \ a_{3M} \ a_{4M}]^T, \ B_M = [b_{1M} \ b_{2M} \ b_{3M} \ b_{4M}]^T; \\ A_R &= [a_{1R} \ a_{2R} \ a_{3R} \ a_{4R}]^T, \ B_R = [b_{1R} \ b_{2R} \ b_{3R} \ b_{4R}]^T; \\ X_L &= [x_1 \ x_2 \ x_3 \ x_4]^T, \ Y_L = [y_1 \ y_2 \ y_3 \ y_4]^T; \\ X_M &= [x_2 \ x_3 \ x_4 \ x_5]^T, \ Y_M = [y_2 \ y_3 \ y_4 \ y_5]^T; \\ X_R &= [x_3 \ x_4 \ x_5 \ x_6]^T, \ Y_R = [y_3 \ y_4 \ y_5 \ y_6]^T; \end{aligned}$$

$$T_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \\ 1 & t_4 & t_4^2 & t_4^3 \end{bmatrix},$$

$$T_M = \begin{bmatrix} 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \\ 1 & t_4 & t_4^2 & t_4^3 \\ 1 & t_5 & t_5^2 & t_5^3 \end{bmatrix},$$

$$T_R = \begin{bmatrix} 1 & t_3 & t_3^2 & t_3^3 \\ 1 & t_4 & t_4^2 & t_4^3 \\ 1 & t_5 & t_5^2 & t_5^3 \\ 1 & t_6 & t_6^2 & t_6^3 \end{bmatrix}.$$

The approximation for the unknown curve between the third and fourth points of the subset makes use of the preliminary cubic polynomials to obtain slopes and the change of the slope. That is, for the third point,

$$\left. \frac{dx}{dt} \right|_{t_3} = \frac{1}{2} \left[\left. \frac{dx_L}{dt} \right|_{t_3} + \left. \frac{dx_M}{dt} \right|_{t_3} \right], \quad (16)$$

$$\left. \frac{dy}{dt} \right|_{t_3} = \frac{1}{2} \left[\left. \frac{dy_L}{dt} \right|_{t_3} + \left. \frac{dy_M}{dt} \right|_{t_3} \right], \quad (17)$$

$$\left. \frac{d^2x}{dt^2} \right|_{t_3} = \frac{1}{2} \left[\left. \frac{d^2x_L}{dt^2} \right|_{t_3} + \left. \frac{d^2x_M}{dt^2} \right|_{t_3} \right], \quad (18)$$

$$\left. \frac{d^2y}{dt^2} \right|_{t_3} = \frac{1}{2} \left[\left. \frac{d^2y_L}{dt^2} \right|_{t_3} + \left. \frac{d^2y_M}{dt^2} \right|_{t_3} \right]. \quad (19)$$

Replacing the left and middle polynomials with the middle and right ones respectively, we obtain the following equations for the fourth point:

$$\left. \frac{dx}{dt} \right|_{t_4} = \frac{1}{2} \left[\left. \frac{dx_M}{dt} \right|_{t_4} + \left. \frac{dx_R}{dt} \right|_{t_4} \right], \quad (20)$$

$$\left. \frac{dy}{dt} \right|_{t_4} = \frac{1}{2} \left[\left. \frac{dy_M}{dt} \right|_{t_4} + \left. \frac{dy_R}{dt} \right|_{t_4} \right], \quad (21)$$

$$\left. \frac{d^2x}{dt^2} \right|_{t_4} = \frac{1}{2} \left[\left. \frac{d^2x_M}{dt^2} \right|_{t_4} + \left. \frac{d^2x_R}{dt^2} \right|_{t_4} \right], \quad (22)$$

$$\left. \frac{d^2y}{dt^2} \right|_{t_4} = \frac{1}{2} \left[\left. \frac{d^2y_M}{dt^2} \right|_{t_4} + \left. \frac{d^2y_R}{dt^2} \right|_{t_4} \right]. \quad (23)$$

Figure 6 shows the process, that is, by generating the left, middle, right curves of 6-point data set, a smooth curve between the third and fourth points is generated.

Repeating the above process by adding one point at the end and deleting the first point in the subset, the middle and right polynomials becomes the left and middle ones for the next subset respectively, so the resulting polynomials between the fourth and fifth points join smoothly with the previously determined curve between the third and fourth points. In another words, the second-order derivatives at any points of the trajectory are continuous.

Since the calculated segment for each local subset is from point 3 to point 4, the first and the last two way-points are

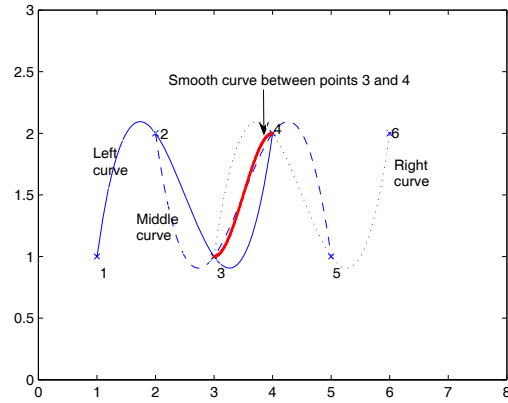


Fig. 6. Generating polynomial approximation of a 6-point data set

not included in the smoothing process. Therefore, 2 new data points, a and b, are added on the straight line connecting the first two points. They are chosen to be very close to the first point and to each other. This will lead to the first point being the third in the first local subset. Similarly, two points, c and d, are added to the end of the curve to make the graph complete. Figure 7 illustrates the idea.

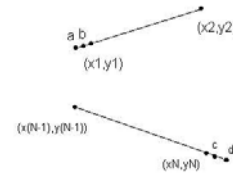


Fig. 7. Pseudo-points are added at the beginning and the end of the way-points.

We summarize the solution to Problem 2 in the following theorem:

Theorem 2: Problem 2 is solved by generating boundary conditions at each way-point according to (16)-(23), and then following the procedure in Theorem 1.

VII. SIMULATION RESULTS

We have performed Matlab simulations for generating global collision-free trajectories. Figure 8 shows a smooth trajectory connecting four way-points (denoted as rectangular in red) between the start and the goal, and avoiding two moving obstacles at the last two path segments. The robot parameters we used are $R = 1, l = 0.8$, and the obstacles' radius is $r_i = 0.5, i = 1, \dots, 4$. The starting positions of the four obstacles, denoted as O_1, O_2, O_3, O_4 in the figure, are $(9, 17), (15, 18), (30, 17), (40, 17)$, respectively.

When the robot is at the third way-point, it detects O_1 and O_2 with constant moving velocities $(v_x, v_y) = (-0.2, 0.32)$ and $(-0.1, 0.2)$ respectively. It plans path 1 and the corresponding control parameters are $(c_6, d_6) = (0, -8.4854 \times 10^{-7})$. When the robot detects the change

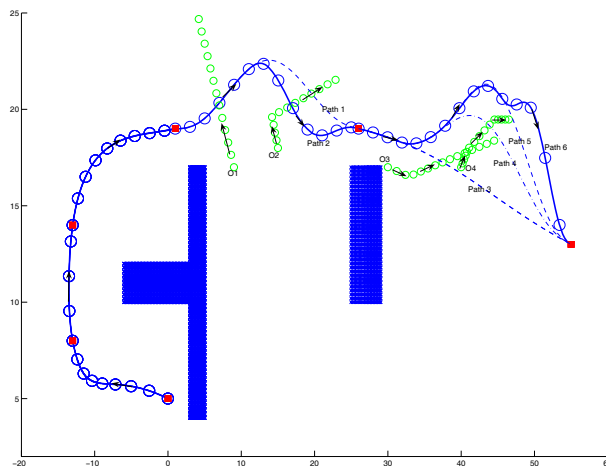


Fig. 8. Global collision-free trajectory. Circles are drawn in equal-time intervals.

of the velocity of $O2$ to $(v_x, x_y) = (0.55, 0.12)$, it re-plan its path to path 2 and $(c_6, d_6) = (0, -1.4269 \times 10^{-5})$ correspondingly. Then, at the fourth way-point, the robot detects $O3$ and $O4$ with velocity $(v_x, x_y) = (0.6, 0.1)$ and $(0.1, 0.1)$, it plans its path as path 3 denoted in the figure with $(c_6, d_6) = (0, 0)$. When it detects the change of the velocity of $O3$ to $(v_x, x_y) = (0.5, 0.08)$, path 4 is planned with $(c_6, d_6) = (0, -1.0015 \times 10^{-6})$. It then detects the velocity change of $O4$ to $(v_x, x_y) = (0.35, 0.15)$, path 5 is planned with $(c_6, d_6) = (0, -3.8644 \times 10^{-6})$. When the second velocity change of $O4$ is detected as $(v_x, x_y) = (0.245, 0)$, path 6 is planned with $(c_6, d_6) = (0, -2.2675 \times 10^{-4})$. we can see that the final path denoted by the solid line in the figure avoids all moving obstacles successfully. The planned path is not optimal due to no performance of the path is taken into consideration in the design. The recalculation of the path parameters c_6 and d_6 is very fast due to their analytic expressions and the robot is able to respond real time.

VIII. CONCLUSIONS

We have studied global trajectory generation for car-like robots in the paper. The global trajectory is composed of regional path segments, which are parametric polynomials. A collision avoidance criterion was developed, and collision avoidance with moving obstacles is achieved by changing parameters of the regional trajectories. The global trajectory is smooth since 1) the regional trajectory is smooth, and 2) the boundary conditions at way-points has second-order continuity. The later one was guaranteed by a smooth irregular curve method developed in the paper. The global path is physically feasible by car-like robots that have nonholonomic constraints. Explicit steering control is constructed in light of differential flatness. Simulations show satisfactory results. Future work includes performance analysis and shortest path generation.

REFERENCES

- [1] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1419–1424, 1986.
- [2] M. Fliess, J. Levine, Ph. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. *Int. J. of Control*, 61(6):1327–1361, 1995.
- [3] Y. Guo, Z. Qu, and J. Wang. A new performance-based motion planner for nonholonomic mobile robots. In *The 3rd Performance Metrics for Intelligent Systems Workshop*, Gaithersburg MD, 2003.
- [4] J. L. Junkins and J. R. Jancaitis. Smooth irregular curves. *Photogrammetric Engineering*, 38(6):565–573, 1972.
- [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] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [7] S. Lavalle and J. Kuffner. Randomized kinodynamic planning. *Int. J. Robot. Res.*, 20:378–400, 2001.
- [8] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: steering using sinusoids. *IEEE Trans. Automat. Contr.*, 38:700–716, 1993.
- [9] W. Nelson. Continuous-curvature paths for autonomous vehicles. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1260–1264, 1989.
- [10] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.*, 17:760–772, 1998.
- [11] A. Piazzzi, 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.
- [12] 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.
- [13] J. A. Reeds and R. A. Shepp. Optimal path for a car that goes both forward and backward. *Pacific J. Math.*, 145:367–393, 1990.
- [14] 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.
- [15] H. J. Sussmann and G. Tang. Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control. Technical Report Tech. Rep. SYSCON-91-10, Rutgers Univ., Piscataway, NJ, 1991.
- [16] 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.