A Reduced-Order Analytical Solution to Mobile Robot Trajectory Generation in the Presence of Moving Obstacles

Jing Wang, Zhihua Qu, Yi Guo and Jian Yang
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
Email: qu@mail.ucf.edu

***Abstract*— This paper addresses the problem of determining a collision-free path for a mobile robot moving in a dynamically changing environment. By explicitly considering kinematic model of the robot, the family of feasible trajectories and their corresponding steering controls are derived in a closed form. Then, a new collision avoidance condition is developed for the dynamically changing environment, it consists of a time criterion and a geometrical criterion, and it has explicit physical meanings in both the transformed space and the original working space. By imposing the avoidance condition, one can determine the corresponding steering angle for collision avoidance in a closed form. Such a path meets all boundary conditions, is continuous, and can be updated in real time once a change in the environment is detected. Simulations show that the proposed method is effective.**

I. Introduction

In the real-world applications, it is desirable that mobile robots are capable of exploring or moving within a dynamic environment. In addition, the environment is usually uncertain as complete information and future trajectories of obstacles cannot be assumed apriori. In this context, the problem naturally arising is how to real-time plan a collision-free path in the presence of dynamically moving objects and with a limited sensing range. A preferred solution to the problem would be one that takes kinematic constraints into consideration, explicitly handles dynamically moving objects, and is analytical.

Standard motion planning approaches [1], such as potential field [2] and vector field histogram [3], are developed to deal with geometrical constraints, more specifically, holonomic systems in the presence of static obstacles. For nonholonomic systems such as mobile robots, their kinematic constraints make time derivatives of some configuration variables non-integrable, and hence a collision-free path in the configuration space is not necessarily feasible (that is, may not achievable by steering controls) [4], [5]. Up to now, most of the existing results deal with nonholonomic systems and object avoidance in one of the two ways. One way is to exclusively focus upon motion planning under nonholonomic constraints. Without considering obstacles, many algorithms have been proposed, for instance, differential geometry [6], differential flatness [7], input parameterization [8], [9], [10], optimal control [11]. The second way is to modify the result from a holonomic planner so the resulting path is feasible. For example, the online suboptimal obstacle avoidance algorithm in [12] is based on Hamilton-Jacobi-Bellman equation [13], [14], it admits stationary obstacles, a planned path is holonomic and its feasibility has to be verified for a chosen nonholonomic mobile robot. The nonholonomic path planner in [15] is based on the same principle, that is, a path is generated by ignoring nonholonomic constraints and it is then made feasible via approximation by using a sequence of such optimal path segments as those in [16].

Exhaustive search or numerical iteration based methods have also been used to deal with nonholonomic constraints and collision avoidance. The search based algorithm [17] involves discretization of the configuration space in order to build and search a graph whose nodes are small axis-parallel cells, two cells are called to be adjacent if there is a feasible path segment between them, and these path segments are constructed by discretizing the controls and integrating the equations of motion. In [18], nonholonomic motion planning is formulated as a nonlinear least squares problem in an augmented space, obstacle avoidance is included as inequality constraints, and a solution is found numerically. In [19], trajectory planning (so called kinodynamic planning) is pursued by considering first-order differential equations and static obstacles and by finding appropriate inputs through a random tree search.

There have been a few results on dealing with moving obstacles. It is proposed in [20] that, if the entire trajectories of the moving obstacles are known apriori, an $(n+1)$ dimensional configuration-time space can be formed by treating the time as an state variable and recasting the dynamic motion planning problem into a static one. In [21], kinodynamic motion planning with moving obstacles is done using a randomizd motion planner in which a control is chosen randomly from the set of admissible values to integrate equation of motion and, if the resulting local trajectory is collision free, its endpoint is put into a probabilistic roadmap. Similar to the approach in [17], the random motion planning of [21] is a search method. In [22], the dynamic motion planning problem is decomposed into two subproblems: a static path planning problem and a velocity planning problem. However, this approach requires complete information (including future trajectories), and its solution is not guaranteed. To the best of our knowledge, there has been no comprehensive result on motion planning for nonholonomic systems operating in a dynamical and uncertain environment.

In our recent work [23], a new analytic solution to mobile robot trajectory generation in the presence of moving obstacles is proposed. Specifically, it has been shown that, for a car-like mobile robot (and others in the $(2, 4)$ chained form), a family of 6th-order piecewise-constant polynomials can be used to describe feasible trajectories (for which steering controls are explicitly found) and that, upon satisfying all boundary conditions, collision-free trajectories can be expressed in terms of one parameter. In this article, we simplify the method in [23] and use a 5th-order piecewise-constant polynomials to parameterize the feasible trajectories. Based on the developed new collision avoidance criterion, we determine the corresponding steering angle so as to generate a collision-free trajectory. The resulting trajectory is continuous, and the corresponding steering controls are piecewise continuous. As a result of the piecewise representations used, the paradigm works if obstacles have varying speeds and if on-board sensors has a limited range.

II. Problem Formulation

In this paper, we shall consider the general problem of trajectory planning for mobile robots in a dynamic and changing environment. As shown in figure 1, possible 2-D environmental changes are due to limited ranges of on-board sensors and to appearance and/or motion of objects. To solve the problem, one can make the following choices without loss of any generality:

- The robot under consideration is represented by a 2-dimensional circle with center at $O(t) = (x, y)$ and of radius r_0. Its motion is controlled but *nonholonomic* and is represented by the velocity vector $v_r(t)$. The range of its sensors is also described by a circle centered at $O(t)$ and of radius R_s.
- The ith object, $i = 1, \cdots, n$, will be represented by a circle centered at point $O_i(t)$ and of radius r_i, denoted by $B_i(O_i(t), r_i)$. For moving objects, the origin $O_i(t)$ is time varying and moving with linear velocity vector $v_i(t)$.
- The robot starts at initial position O_o and initial orientation θ_0, moves collision free, and arrives at final position O_f and with final orientation θ_f.

Intuitively, the trajectory planning problem has at least one solution if the robot is capable of moving sufficiently fast and if there exists a finite time instant $T_f > 0$ such that the free space is connected and $O_f \notin B_i(O_i(t), r_i)$ for $t \geq t_0 + T_f$ and for all $i = 1, \cdots, n_o$.

However, the general trajectory planning problem is physically ill-posed as its solution will require apriori knowledge of both the objects' present and future motion information. To overcome this difficulty while making the proposed method

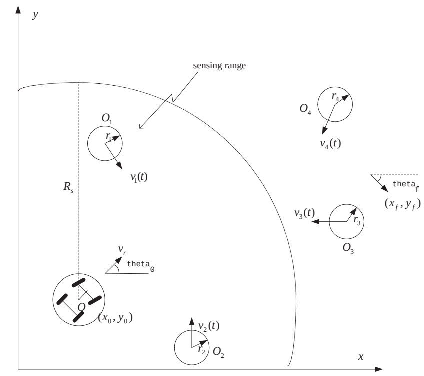

Fig. 1. A general setting of trajectory planning in the presence of moving obstacles

practically implementable, we use piecewise constants and functions to represent arbitrary functions. Specifically, within a specified period of time $t \in [t_0 + kT_s, t_0 + (k+1)T_s)$ (where T_s is often small),

- Velocity v_i of the ith object is constant, denoted by v_i^k.
- Only the objects in the range of sensors are considered.
- Trajectory and control of the robot are chosen to be functions with piecewise constant parameters.

In some applications, not only is the sensor range limited, the final position O_f may not be fixed either and thus can also be represented by a piecewise constant function. Therefore, trajectory planning or re-planning is done for a snapshot of figure 1, and is constantly updated. To do so efficiently online, the proposed piecewise-constant parameterization must yield analytical solutions.

A. Robot Modeling

In this paper, a new paradigm is proposed to plan trajectories and avoid moving obstacles for nonholonomic mobile robots. In the new paradigm, kinematic models of the car-like robots are explicitly considered in trajectory planning.

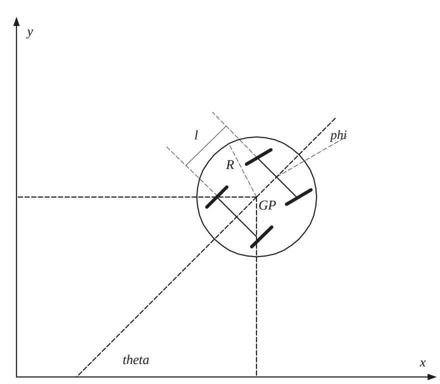

Fig. 2. A car-like robot

The car-like robot is shown in figure 2, its front wheels are steering wheels, and its rear wheels are driving wheels but have a fixed orientation. The distance between the two wheel-axle centers is l, the midpoint along the line connecting the axle centers is set to be the guidepoint (GP), and the whole vehicle is physically within a circle of radius R and centered at the guidepoint. Trajectory planning will be done for the guidepoint. Let the generalized coordinates be $q = [x\ y\ \theta\ \phi]^T$, where (x, y) are the Cartesian coordinates of the guidepoint, θ is the orientation of the robot body with respect to the x axis (that is, the slope angle of the line passing through the guidepoint and center of the back axle), ϕ is the steering angle, and let ρ be the radius of the (back) driving wheels. Its kinematic model can be given by:

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

where u_1 is the angular velocity of the driving wheels, and u_2 is the steering rate of the (front) guiding wheels. The kinematic model (1) has singularity at $\phi = \pm\pi/2$, which does not occur mathematically or in practice by limiting the range of ϕ within $(-\pi/2\ \pi/2)$. The range of θ is also set within $(-\pi/2\ \pi/2)$ to ensure an one-to-one mapping of following transformations of coordinates and inputs:

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

and

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

Under the transformations, kinematic model (1) can be mapped into the so-called chained form, that is,

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \\ \dot{z}_4 &= z_3 v_1 \end{aligned} \tag{4}$$

In this paper, we use the car-like robot as the example and adopt the chained form in solving the problem of trajectory planning. The proposed steering paradigm for trajectory planning and object avoidance has the following features:

- Kinematic models (and possibly dynamic model) of robots are explicitly considered.
- Motion of objects are represented by piecewise constant velocities, and collision avoidance criterion is defined analytically and thus less conservative than the existing methods.
- Piecewise constant parameterization will be used to define trajectory and steering control, and their solutions are obtained in closed-form.

III. Proposed Steering Paradigm

The proposed paradigm consists of three basic steps, and it is based on the two corner stones of steering and collision-free criterion (newly defined for moving objects). On one side, it begins with kinematic model, that is, steering strategies are used to find out the class of physically achievable trajectories. On the other hand, collision avoidance criterion can be explicitly developed for moving objects. As the third step, a specific class within all achievable trajectories will first be parameterized and then solved using the object avoidance criterion.

A. Feasible Trajectories

A trajectory is *feasible* if it satisfies both the boundary conditions imposed and dynamics of the kinematic model (if it exists). The chained form in equation (4) is used as the standard one to study and determine trajectories that observe the kinematic model. The following result shows a general class of feasible trajectories in terms of transformed state z. The proof can be done by direct computation.

Lemma 1: : Consider the kinematic model in chained form (4). Then, given any boundary conditions $z(t_0) = z^0 = [z_1^0, z_2^0, z_3^0, z_4^0]^T$ and $z(t_f) = z^f = [z_1^f, z_2^f, z_3^f, z_4^f]^T$ (for some $t_f > 0$), there exist inputs $v_1 = C$ (for some non-zero constant C) and v_2 to make any trajectory $z_4 = F(z_1)$ (in the $z_1 - z_4$ plane) feasible provided that $z_1^0 \neq z_1^f$ and that, through transformation (2), function $z_4 = F(z_1)$ also satisfies all the boundary conditions in the original state space.

Remark 3.1: If $z_1^f = z_1^0$, $v_1 = 0$ which causes a singularity in determining v_2. In this case, the singularity can be avoided by choosing an intermediate point z^m with $z_1^m \neq z_1^0 = z_1^f$ and by proceeding with planning two paths in the $z_1 - z_4$ plane. $\diamondsuit$

Lemma 1 shows that, by making $z_4 = F(z_1)$ conform the boundary conditions $(x_0, y_0, \theta_0, \phi_0)$ and $(x_f, y_f, \theta_f, \phi_f)$ in the original state space, the steering problem can be solved. In this paper, for a feasible trajectory, the following boundary conditions on boundary points, slopes, and curvatures are applied: given $z_4 = F(z_1)$,

$$\begin{aligned} &z_1^0 = x_0 - \tfrac{l}{2}\cos(\theta_0), \quad F(z_1^0) = y_0 - \tfrac{l}{2}\sin(\theta_0), \\ &\tfrac{dz_4}{dz_1}\Big|_{z_1 = z_1^0} = \tan(\theta_0), \quad \tfrac{d^2 z_4}{d(z_1)^2}\Big|_{z_1 = z_1^0} = \tfrac{\tan(\phi_0)}{l\cos^3(\theta_0)}, \end{aligned} \tag{5}$$

$$\begin{aligned} &z_1^f = x_f - \tfrac{l}{2}\cos(\theta_f), \quad F(z_1^f) = y_f - \tfrac{l}{2}\sin(\theta_f), \\ &\tfrac{dz_4}{dz_1}\Big|_{z_1 = z_1^f} = \tan(\theta_f), \quad \tfrac{d^2 z_4}{d(z_1)^2}\Big|_{z_1 = z_1^f} = \tfrac{\tan(\phi_f)}{l\cos^3(\theta_f)}. \end{aligned} \tag{6}$$

Remark 3.2: The above boundary conditions on the second-order derivatives, together with those on first-order derivatives, are equivalent to boundary curvatures (κ) of the trajectory as $\kappa = \frac{d^2 z_4}{dz_1^2} / \left[1 + \left(\frac{dz_4}{dz_1}\right)^2\right]^{3/2}$. $\diamondsuit$

B. Criterion for Avoiding Dynamic Objects

To illustrate the criterion in the proposed steering paradigm, consider the robot (of coordinates $(x(t), y(t))$) and the ith object (of coordinates $(x_i(t), y_i(t))$) in figure 3 for the period

$t \in [t_0 + kT_s, t_0 + (k+1)T_s)$. In the figure, the robot is moving at a vector velocity $v_r \triangleq [\ \dot{x}(t)\ \dot{y}(t)\ ]^T$ (which is to be determined), the object has an initial location $O_i = (x_i^k, y_i^k)$ where $x_i^k = x_i(t_0 + kT_s)$ and $y_i^k = y_i(t_0 + kT_s)$, and point O_i is moving at a known constant velocity $v_i^k \triangleq [\ v_{i,x}^k\ v_{i,y}^k\ ]^T$.

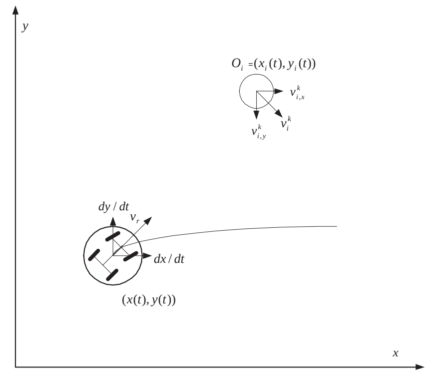

Fig. 3. Steering paradigm: robot and the ith object

To develop a criterion for collision avoidance, we define the robot velocity relative to that of the ith object as

$$v_{r,i}^k \triangleq v_r - v_i^k = \begin{bmatrix} v_{r,i,x}^k \\ v_{r,i,y}^k \end{bmatrix} = \begin{bmatrix} \dot{x} - v_{i,x}^k \\ \dot{y} - v_{i,y}^k \end{bmatrix}. \tag{7}$$

Using the relative velocity, figure 3 is transformed into figure 4 in which the object is "static." According to figure 4, the collision avoidance criterion in the $y - x$ plane should be: for $x_i' \in [\underline{x}_i',\ \overline{x}_i']$ with $\underline{x}_i' = x_i^k - r_i - R$ and $\overline{x}_i' = x_i^k + r_i + R$,

$$(y_i' - y_i^k)^2 + (x_i' - x_i^k)^2 \geq (r_i + R)^2,$$

where $x_i' = x - v_{i,x}^k \tau$, $y_i' = y - v_{i,y}^k \tau$, and $\tau = t - (t_0 + kT_s)$ for $t \in [t_0 + kT_s, t_0 + T]$, with T being the time for the mobile robot to complete its maneuver.

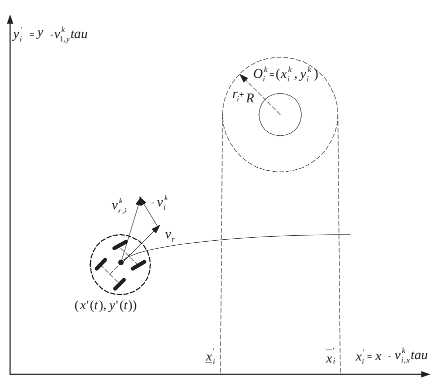

Fig. 4. Relative velocity of the robot with respect to the ith obstacle

It follows from state transformation (2) that, given any steerable path $z_4 = F(z_1)$, the corresponding feasible path in the $x - y$ plane is

$$y = F(x - 0.5l\cos(\theta)) + 0.5l\sin(\theta).$$

Thus, the corresponding collision avoidance criterion in the transformed $z_4 - z_1$ space is: whenever $x_i^k \in [z_{1,i}' + 0.5l\cos(\theta) - r_i - R,\ z_{1,i}' + 0.5l\cos(\theta) + r_i + R]$,

$$\left(z_{4,i}' + \frac{l}{2}\sin(\theta) - y_i^k\right)^2 + \left(z_{1,i}' + \frac{l}{2}\cos(\theta) - x_i^k\right)^2 \geq (r_i + R)^2, \tag{8}$$

where $z_{1,i}' = z_1 - v_{i,x}^k \tau$ and $z_{4,i}' = z_4 - v_{i,y}^k \tau$.

Note that, although θ can be determined from z_3 and z_3 can be obtained as a result of applying lemma 1, exact mapping from z to (x, y, θ) should not be used to numerically solve the problem of trajectory planning by imposing criterion (8). Instead, we choose to develop a new criterion only in terms of z_1 and z_4 (or z_1' and z_4') so that analytical solution can be found for the problem of trajectory planning. To this end, note that all possible locations of point $(z_{1,i}', z_{4,i}')$ are on the left semi circle centered at (x_i^k, y_i^k) and of radius $l/2$ for $\theta \in [-\pi/2, \pi/2]$. As shown in figure 5, plotting a family of circles of radius $(r_i + R)$ along the left semi circle renders the region from which the robot must stay clear, and the region is completely covered by the *unshaded* portion of the circle centered at (x_i^k, y_i^k) and of radius $(r_i + R + l/2)$. Mathematically, the proposed collision avoidance criterion in the $z_4 - z_1$ plane is:

$$(z_{4,i}' - y_i^k)^2 + (z_{1,i}' - x_i^k)^2 \geq \left(r_i + R + \frac{l}{2}\right)^2, \tag{9}$$

provided that

$$z_{1,i}' \in [x_i^k - r_i - R - 0.5l,\ x_i^k + r_i + R]. \tag{10}$$

It is apparent from figure 5 that criterion (9) implies criterion (8). Once a steering method is chosen, the time interval during which criterion (9) should be imposed to avoid collision can be found from (10). That is, the proposed collision avoidance scheme has two parts: time criterion (10), and geometrical criterion (9).

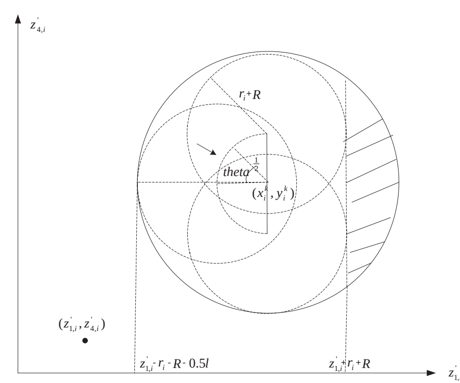

Fig. 5. Illustration of the collision avoidance criterion in the transformed plane

C. A Solution to Feasible Collision-Free Trajectory

Let T_s be the sampling period such that $\overline{k} = T/T_s$ is an integer, then for $k = 0, \cdots, \overline{k} - 1$, a specific candidate class of feasible trajectories are parameterized by a quintic polynomial function as

$$z_4(z_1) = F(z_1) = a^k f(z_1), \tag{11}$$

where $a^k = [a_0^k, a_1^k, \cdots, a_5^k]$ is a constant vector to be determined, and $f(z_1) = [1, z_1(t), (z_1(t))^2, \cdots, (z_1(t))^5]^T$ is the vector composed of basis functions of $z_1(t)$. Let the steering control $v_1(t)$ be

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

then, for $k > 0$, we can obtain the boundary condition at time instant $t = t_0 + kT_s$ as:

$$\begin{aligned}
z_1^k &= z_1^0 + \frac{k(z_1^f - z_1^0)}{\bar{k}}, \\
z_2^{k^-} &= z_2^{k-1} + \int_{t_0+(k-1)T_s}^{t_0+kT_s} v_2^{k-1}(t)dt, \\
z_3^k &= z_3^{k-1} + \frac{z_1^f - z_1^0}{\bar{k}} z_2^{k-1} \\
&\quad + \frac{z_1^f - z_1^0}{T} \int_{t_0+(k-1)T_s}^{t_0+kT_s} \int_{t_0+(k-1)T_s}^{s} v_2^{k-1}(t)dtds, \\
z_4^k &= z_4^{k-1} + \frac{z_1^f - z_1^0}{\bar{k}} z_3^{k-1} + \frac{T_s}{2} \frac{z_1^f - z_1^0}{\bar{k}} z_2^{k-1} \\
&\quad + \int_{t_0+(k-1)T_s}^{t_0+kT_s} \int_{t_0+(k-1)T_s}^{\tau} \int_{t_0+(k-1)T_s}^{s} C v_2^{k-1}(t)dtdsd\tau.
\end{aligned} \tag{13}$$

Remark 3.3: The notation for boundary condition $z_2^{k^-}$ has been used to express the computed value of z_2 at time instant $t = t_0 + kT_s$ based on the steering control $v_2^{k-1}(t)$ for $t \in [t_0+(k-1)T_s, t_0+kT_s]$. At time instant $t = t_0+kT_s$, we will determine whether the boundary condition $z_2^{k^-}$ will satisfy the developed obstacle avoidance criterion. If not, we will change the value of $z_2^{k^-}$ to $z_2^{k^+}$ at time instant $t = t_0 + kT_s$ which is computed out using the obstacle avoidance condition (will be clearly shown shortly), and then according to new established boundary conditions $z_2^{k^+}$, z_3^k and z_4^k, the coefficients a^k and the corresponding steering control $v_2^k(t)$ for the time interval $t \in [t_0 + kT_s, t_0 + (k+1)T_s]$ will be determined.

To this end, the coefficients can be obtained by applying boundary conditions either (5) or (13) and (6) as:

$$(a^k)^T = (B^k)^{-1} Y^k, \tag{14}$$

where

$$Y^k = \begin{bmatrix} z_4^k \\ z_3^k \\ z_2^k \\ y_f - \frac{l}{2}\sin(\theta_f) \\ \tan(\theta_f) \\ \frac{\tan(\phi_f)}{l\cos^3(\theta_f)} \end{bmatrix},$$

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

and z_2^k takes the value of $z_2^{k^-}$ or $z_2^{k^+}$ based on obstacle avoidance condition.

By (11), since $z_4(z_1) = f^T(z_1)a^k = f^T(z_1)(B^k)^{-1}Y^k$, noting the expression of Y^k, we can further have

$$z_4(z_1) = f_1(z_1)z_2^k + f_2(z_1), \tag{15}$$

where $f_1(z_1)$ and $f_2(z_1)$ are functions depending on $z_1(t)$ and boundary conditions $z_4^k, z_3^k, z_4^f, z_3^f, z_2^f, z_1^k$ and z_1^f as follows:

$$\begin{aligned}
f_1(z_1) &= f^T(z_1)(B^k)^{-1} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \\
f_2(z_1) &= f^T(z_1)(B^k)^{-1} \begin{bmatrix} z_4^k & z_3^k & 0 & z_4^f & z_3^f & z_2^f \end{bmatrix}^T.
\end{aligned}$$

Let the centers of objects O_i be located at (x_i^k, y_i^k) at $t = t_0 + kT_s$, and that these objects are all moving with known constant velocities $v_i^k \triangleq [\, v_{i,x}^k \; v_{i,y}^k \,]^T$ for $t \in [t_0 + kT_s, t_0 + (k+1)T_s)]$. To satisfy the obstacle avoidance criterion (9), we will determine the value of z_2^k based on the following second-order inequality (or inequalities): $\forall i \in \{1, \cdots, n_o^k\}$, where n_o^k is the number of obstacles that are in the sensor range of robot at time instant $t_0 + kT_s$,

$$\begin{aligned}
\min_{t\in[\underline{t}_i^*, \bar{t}_i^*]} & g_2(z_1(t), k)(z_2^k)^2 + g_{1,i}(z_1(t), k, \tau)z_2^k + \\
& g_{0,i}(z_1(t), k, \tau)|_{\tau=t-t_0-kT_s} \geq 0,
\end{aligned} \tag{16}$$

where $[\underline{t}_i^*, \bar{t}_i^*] \subset [t_0+kT_s,\ T]$ is the time interval during which

$$x_i^k \in [z_1(t) - v_{i,x}^k\tau - r_i - R,\ z_1(t) - v_{i,x}^k\tau + 0.5l + r_i + R]. \tag{17}$$

In (16), functions $g_2(\cdot)$, $g_{1,i}(\cdot)$ and $g_{0,i}(\cdot)$ are defined as follows:

$$\begin{aligned}
g_2(z_1(t), k) &= [f_1(z_1)]^2, \\
g_{1,i}(z_1(t), k, \tau) &= 2f_1(z_1)\left[f_2(z_1) - y_i^k - v_{i,y}^k\tau\right], \\
g_{0,i}(z_1(t), k, \tau) &= \left[f_2(z_1) - y_i^k - v_{i,y}^k\tau\right]^2 \\
&\quad + (z_1(t) - x_i^k - v_{i,x}^k\tau)^2 - (r_i + R + 0.5l)^2.
\end{aligned}$$

with

$$x_i^0 = x_i(t_0), \quad y_i^0 = y_i(t_0); \quad x_i^k = x_i^0 + T_s\sum_{j=0}^{k-1} v_{i,x}^j,$$

$$y_i^k = y_i^0 + T_s\sum_{j=0}^{k-1} v_{i,y}^j, \quad \text{if } k > 0.$$

If $z_2^{k^-}$ in (13) satisfies (16), we will let $z_2^k = z_2^{k^-}$, otherwise, we pick up a $z_2^{k^+}$ according to (16) as the value of z_2^k. It is obvious that by adjusting the boundary condition z_2^k, i.e., the corresponding steering angle ϕ_k, and the curve shape will be changed to evade the obstacles.

Once we obtain a^k using (14), correspondingly, the steering input $v_2^k(t)$ for $t \in [t_0 + kT_s, t_0, (k+1)T_s]$ can be determined as follows: let

$$v_2^k(t) = C_0 + C_1(t - t_0 - kT_s) + C_2(t - t_0 - kT_s)^2$$

where $C_j, j = 0, 1, 2, 3$ are constants. Directly integrating (4) yields

$$\begin{aligned} z_1(t) &= z_1^k + C(t - t_0 - kT_s) \\ z_2(t) &= z_2^k + C_0(t - t_0 - kT_s) + \frac{C_1}{2}(t - t_0 - kT_s)^2 \\ &\quad + \frac{C_2}{3}(t - t_0 - kT_s)^3 \\ z_3(t) &= z_3^k + C z_2^k (t - t_0 - kT_s) + \frac{CC_0}{2}(t - t_0 - kT_s)^2 \\ &\quad + \frac{CC_1}{6}(t - t_0 - kT_s)^3 + \frac{CC_2}{12}(t - t_0 - kT_s)^4 \\ z_4(t) &= z_4^k + C z_3^k (t - t_0 - kT_s) + \frac{C^2 z_2^k}{2}(t - t_0 - kT_s)^2 \\ &\quad + \frac{C^2 C_0}{6}(t - t_0 - kT_s)^3 + \frac{C^2 C_1}{24}(t - t_0 - kT_s)^4 \\ &\quad + \frac{C^2 C_2}{60}(t - t_0 - kT_s)^5 \end{aligned} \tag{18}$$

On the other hand, substituting $z_1(t) = z_1^k + C(t - t_0 - kT_s)$ into $z_4 = a^k f(z_1)$ yields

$$\begin{aligned} z_4(t) &= b_0 + b_1(t - t_0 - kT_s) + b_2(t - t_0 - kT_s)^2 \\ &\quad + b_3(t - t_0 - kT_s)^3 + b_4(t - t_0 - kT_s)^4 \\ &\quad + b_5(t - t_0 - kT_s)^5, \end{aligned} \tag{19}$$

where $b_0 = \sum_{i=0}^5 a_i^k (z_1^k)^i$, $b_1 = a_1^k C + 2a_2^k C z_1^k + 3a_3^k C (z_1^k)^2 + 4a_4^k C (z_1^k)^3 + 5a_5^k C (z_1^k)^4$, $b_2 = a_2^k C^2 + 3a_3^k C^2 z_1^k + 6a_4^k C^2 (z_1^k)^2 + 10a_5^k C^2 (z_1^k)^3$, $b_3 = a_3^k C^3 + 4a_4^k C^3 z_1^k + 10a_5^k C^3 (z_1^k)^2$, $b_4 = a_4^k C^4 + 5a_5^k C^4 z_1^k$, and $b_5 = a_5^k C^5$. Then, in light of (14), we can solve for constants C_i in $v_2^k(t)$ by comparing expressions (19) and (18). The result renders the steering inputs as:

$$\begin{aligned} v_2^k(t) &= 6[a_3^k + 4a_4^k z_1^k + 10a_5^k (z_1^k)^2]C \\ &\quad + 24[a_4^k + 5a_5^k z_1^k](t - t_0 - kT_s)C^2 \\ &\quad + 60a_5^k (t - t_0 - kT_s)^2 C^3. \end{aligned} \tag{20}$$

Following the above derivations, we can obtain the following theorem, the main result of the paper, and it provides an analytical solution to the problem of finding a feasible collision-free trajectory.

Theorem 1: : Consider a nonholonomic car-like robot of (1) and operating in the presence of circular moving obstacles that are centered at O_i and of radius r_i. Then, for any given boundary conditions $q^0 = [x_0, y_0, \theta_0, \phi_0]^T$ and $q^f = [x_f, y_f, \theta_f, \phi_f]^T$ with $\phi_0 = \phi_f = 0$, as defined by (5) and (6), and satisfying the conditions that $x_0 - \frac{l}{2}\sin(\theta_0) \neq x_f - \frac{l}{2}\sin(\theta_f)$ and that $|\theta_0 - \theta_f| < \pi$, a collision-free path can be generated analytically by undertaking the following steps:

(i) For $k = 0, \cdots, \overline{k} - 1$, determine recursively constants z_2^k by ensuring the following second-order inequality (or inequalities) (16)

(ii) A feasible, collision-free path of form (11). in the transformed state is found by solving a^k according to (14).

(iii) The steering inputs to achieve path (11) are given by (12) and (20), for $t \in (t_0 + kT_s, t_0 + (k+1)T_s]$.

(iv) The corresponding feasible, collision-free Cartesian trajectory is given by $y = F(x - 0.5l\cos(\theta)) + 0.5l\sin(\theta)$, where θ can be found in closed form from state transformation (2) under steering inputs (12) and (20) and control mapping (3).

IV. Simulation

In this section, the proposed steering algorithm is simulated to illustrate its effectiveness. In the simulations, the following settings are used:

- Robot parameters: $R = 1$, $l = 0.8$ and $\rho = 0.2$.
- Boundary conditions: $q^0 = (0, 0, \frac{\pi}{4}, 0)$ and $q^f = (17, 10, -\frac{\pi}{4}, 0)$.
- Moving obstacles: $n_0 = 3$,

  $$O_1(t_0) = [5, 0]^T, \quad O_2(t_0) = [9, 4]^T, \quad O_3(t_0) = [19, 10]^T$$

  and $r_i = 0.5$ for $i = 1, 2, 3$.
- Design parameters: $t_0 = 0$, $T = 40$ seconds, and $T_s = 10$ seconds.
- Speeds of obstacles:

  $$v_1^0 = [0, 0.4]^T, v_1^1 = [0.5, 0.2]^T, v_1^2 = v_1^3 = [0.2, 0.2]^T,$$
  $$v_2^0 = [-0.5, 0]^T, v_2^1 = [0.6, 0.1]^T, v_2^2 = v_2^3 = [0.6, 0.1]^T,$$
  $$v_3^0 = [-0.2, -0.1]^T, v_3^1 = [-0.2, 0.1]^T, v_3^2 = v_3^3 = [-0.1, 0.1]^T.$$

  All quantities conform to a given unit system, for instance, meter, meter per second, etc.

In the simulation, the robot's sensor has a limited range $R_s = 7$ so the robot detects the presence of objects 1, 2 and 3 intermittently. Sampling period T_s is chosen to account for speed changes of objects detected, and n_o^k is introduced to account for emergence and disappearance of various objects in the sensing range of the robot. It is obvious that, when T_s elapses or n_o^k increases, the proposed algorithm needs to be applied to update the trajectory and its corresponding steering controls. The evolution of planned trajectories are plotted in figures 6 and 7. Specifically, at $t = 0$, $n_o^0 = 1$ (object 1 only) and the corresponding trajectory is shown by path 1 in figure 6, and it is kept until either T_s elapses or n_o^k changes. At time instant $t = 2.8$, object 2 is detected by the robot sensor, and n_o^0 becomes 2 (objects 1 and 2). Accordingly, using the proposed algorithm and at $t = 2.8$, the value of z_2 (correspondingly the value of ϕ) is updated instantaneously, and for $t \in [2.8, 10]$ the robot will be commanded to follow the path 2 given in figure 6. It is clear from figure 6 that, if robot followed path 1 for the entire time interval $t \in [0, 10]$, a collision between the robot and object 2 will occur around $t = 8$ second. The rest evolution of planned trajectories is conceptually the same. In particular, at $t = 10$, we have $n_o^1 = 2$(objects 1 and 2) and hence the trajectory is kept. Within the interval $t \in [10, 20)$, objects 1 and 2 gradually move out of the sensing range, nonetheless the robot trajectory can remain to be path 2 in figure 6 (or, one could choose to replan the trajectory). At $t = 20$, $n_o^2 = 0$ (no object), the algorithm chooses to change z_2 (ϕ) again so that robot will follow follow path 3. Around $t = 25$, objects 1 and 3 are detected, n_o^2 is updated to be $n_o^2 = 2$, and z_2 is updated again by the algorithm and the robot follows path 4. At $t = 30$, $n_o^3 = 1$ (object 3), the z_2 is changed again and the robot is commanded to follow path 5. The variation of ϕ is shown in figure 7.

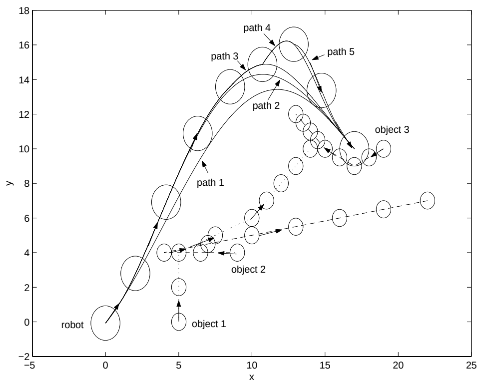

Fig. 6. Collision-free path of robot (solid line), obstacle 1 (dotted line), obstacle 2 (dashdot line) and obstacle 3 (dashed line)

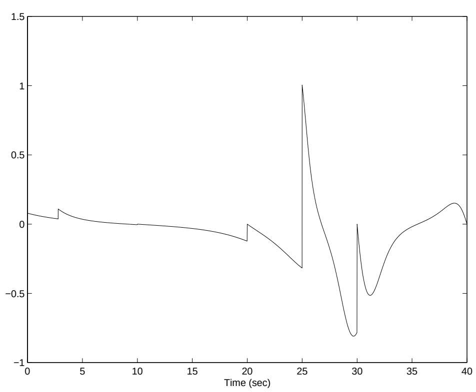

Fig. 7. Entire trajectory of $\phi(t)$

V. Conclusion

In this paper, a new collision avoidance paradigm is proposed to solve the problem of real-time trajectory generation. While the robotic platform is chosen to be a 4-wheel car-like mobile vehicle, the proposed paradigm uses the chained form as the basic model and therefore is applicable to other nonholonomic systems. Based on a piecewise constant polynomial parameterization of all feasible trajectories, the proposed scheme prevents any collision by checking a time criterion and then a geometrical criterion, and it yields analytical solutions to collision-free path(s) and the corresponding steering controls. The piecewise constant representation of feasible trajectories and steering controls enables the proposed method to admit such changes in a dynamical environment as speed change of obstacles, limited sensor range (and the corresponding appearance and disappearance of obstacles), and resetting of terminal conditions.

References

[1] J. Latombe, *Robot Motion Planning*, Kluwer Academic Publisher, 1998, Boston.

[2] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robotic. Autom.*, vol. 8, pp. 501–518, 1992.

[3] J. Borenstein and Y. Koren, "The vector field histogram — fast obstacle avoidance for mobile robots," *IEEE Trans. Robotic. Autom.*, vol. 7, pp. 278–288, 1991.

[4] J.-P. Laumond, *Robot Motion Planning and Control*, Springer-Verlag, 1998, London.

[5] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Inc., 1994, Boca Raton.

[6] H. J. Sussmann and W. Liu, "Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories," in *Tech. Rep. SYSCON-91-02*, Rutgers Ctr. Syst. and Contr., Piscataway, NJ, Feb. 1991.

[7] M. Fliess, J. Levine, Ph. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.

[8] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *IEEE Trans. on Automatic Control*, vol. 38, pp. 700–716, 1993.

[9] S. Monaco and D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," in *Proceedings of the 31st Confer. on Decision and Control*, Tucson, Arizona, Dec. 1992, pp. 1780–1785.

[10] D. Tilbury, R. M. Murray, and S. S. Sastry, "Trajectory generation for the n-trailer problem using goursat normal form," *IEEE Trans. on Automatic Control*, vol. 40, pp. 802–819, 1995.

[11] C. Fernandes, L. Gurvits, and Z. Li, "Near-optimal nonholonomic motion planning for a system of coupled rigid bodies," *IEEE Trans. on Automatic Control*, vol. 39, pp. 450–463, 1994.

[12] S. Sundar and Z. Shiller, "Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 305–310, 1997.

[13] Bryson A.E. and Y.-C. Ho, *Applied Optimal Control*, 2nd Edition, Hemisphere Publishing Corporation, New York, 1975.

[14] Z. Qu and J. R. Cloutier, "A new suboptimal control design for cascaded nonlinear systems," *Optimal Control: Applications and Methods*, vol. 23, pp. 303–328, 2002.

[15] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 10, pp. 577–593, 1994.

[16] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific J. Mathematics*, vol. 145, pp. 367–393, 1990.

[17] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles," in *Proceedings of the 1991 IEEE Int. Confer. on Robotics and Automation*, Sacramento, CA, April 1991, pp. 2328–2335.

[18] A. W. Divelbiss and J. T. Wen, "A path space approach to nonholonomic motion planning in the presence of obstacles," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 443–451, 1997.

[19] S. Lavalle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. of Robotics Research*, vol. 20, pp. 378–400, 2001.

[20] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," in *IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 1986, pp. 1419–1424.

[21] D. Hsu, R. Kindel, J-C.Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. of Robotics Research*, vol. 21, pp. 233–255, 2002.

[22] K. Kant and S. W. Zucker, "Planning collision free trajectories in time-varying environments: a two level hierarchy," in *IEEE International Conference on Robotics and Automation*, Raleigh, NC, 1988, pp. 1644–1649.

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