A New Analytical Solution to Mobile Robot Trajectory Generation in the Presence of Moving Obstacles

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

Clinton E. Plaisted
Lockheed Martin Missiles and Fire Control
5600 Sand Lake Rd. MP-450
Orlando, FL 32819

Abstract—The problem of determining a collision-free path for a mobile robot moving in a dynamically changing environment is addressed in this paper. By explicitly considering kinematic model of the robot, the family of feasible trajectories and their corresponding steering controls are derived in a closed form and also expressed in terms of one adjustable parameter for the purpose of collision avoidance. 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 one (or a class of) collision-free path(s) in a closed form. Such a path meets all boundary conditions, is twice differentiable, and can be updated in real time once a change in the environment is detected. Solvability condition of the problem is explicitly found. And, simulations show that the proposed method is effective.

I. INTRODUCTION

For most 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 a priori. 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]. In particular, the nonholonomic motion planning problem can be recast as an optimal control problem, the Pontryagin’s Maximum Principle can be applied, and it is shown in [12] (improved later in [13]) that the feasible shortest path for a point robot under two boundary conditions is a concatenation of simple pieces (such as an arc and a straight line segment) that belong to 46 three-parameter families of controls. 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 [14] is based on Hamilton-Jacobi-Bellman equation [15], [16], 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 [17] 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 [12]. Exhaustive search or numerical iteration based methods have also been used to deal with nonholonomic constraints and collision avoidance. The search based algorithm [18] 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 [19], 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.

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], the dynamic motion planning problem is decomposed into two subproblems: a static path planning problem and a velocity planning problem. The static
path planning problem is to find a path that avoids all static obstacles, and the velocity planning problem is to determine the velocity of the robot along that path so that there will be no collision. However, this approach requires complete information (including future trajectories), and its solution is not guaranteed. For obstacles moving with known constant velocities, velocity planning can be done using the velocity obstacle concept in [22]. That is, collision does not occur if the robot velocity is chosen such that its velocities relative to the obstacles’ motion do not enter the corresponding collision cones. 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 this article, a new collision avoidance method is proposed to analytically solve the problem of real-time trajectory planning and replanning for nonholonomic mobile robots operating in an environment of multiple dynamically moving obstacles. The proposed method is a three-leg paradigm: mapping the system dynamics into one of chained forms, parameterizing all feasible trajectories by a family of piecewise constant polynomials and determining the corresponding steering controls, and developing a new collision avoidance condition (which consists of a time criterion and a geometrical criterion and is explicit in both the transformed space and the original working space). 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. This parameterization makes it possible to analytically solve for collision-free path(s) by invoking the proposed collision avoidance condition. The resulting trajectory is twice differentiable, 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. It is shown that, so long as collision does not occur at the boundary conditions, the dynamic path generation problem is always solvable and that solutions are given in closed form.

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 of 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 \( R \).
- The \( i \)th object, \( i = 1, \ldots, n \), 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 \( O_0 \) and initial orientation \( \theta_0 \), moves collision free, and arrives at final position \( O_f \) and with final orientation \( \theta_f \).

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 \( i = 1, \ldots, n_o \).

![Fig. 1. A general setting of trajectory planning in the presence of moving obstacles](image)

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 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 \( i \)th 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 application, 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 model of the robots are explicitly considered in trajectory planning, and their dynamic models could also be included (though the latter will not be
considered in this paper). To this end, a kinematic model of a car-like mobile robot is used in this paper to develop a trajectory planning algorithm using the paradigm.

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 \( R \) 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, \( \theta \) 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), and \( \phi \) is the steering angle.

It is assumed that, during normal operation under steering control, the robot rolls without slipping. In terms of motion of two sets of wheels, this translates into the following constraint equations:

\[
\begin{align*}
v^f_x \sin(\theta + \phi) - v^f_y \cos(\theta + \phi) &= 0, \\
v^b_x \sin(\theta) - v^b_y \cos(\theta) &= 0,
\end{align*}
\]

where \((v^f_x, v^f_y)\) and \((v^b_x, v^b_y)\) are \(x\)-axis and \(y\)-axis velocities of the front wheels and the back wheels, respectively. Equations (1) and (2) are non-integrable, and thus belong to the so-called nonholonomic constraints. Let \( \rho \) be the radius of the (back) driving wheels, \( u_1 \) be the angular velocity of the driving wheels, and \( u_2 \) be the steering rate of the (front) guiding wheels. One can express nonholonomic constraints (1) and (2) by the following kinematic model for the car-like robot:

\[
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta} \\
\dot{\phi}
\end{pmatrix} =
\begin{pmatrix}
\rho \cos(\theta) - \frac{\rho}{l} \tan(\phi) \sin(\theta) & 0 \\
\rho \sin(\theta) + \frac{\rho}{l} \tan(\phi) \cos(\theta) & 0 \\
\frac{\rho}{l} \tan(\phi) & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2
\end{pmatrix}.
\]

(3)

Kinematic model (3) has singularity at \( \phi = \pm \pi/2 \), which does not occur mathematically or in practice by limiting the range of \( \phi \) within \((-\pi/2, \pi/2)\). Within this range, the following transformations of coordinates and inputs are well defined:

\[
\begin{align*}
z_1 &= x - \frac{l}{2} \cos(\theta), \\
z_2 &= \frac{\tan(\phi)}{l \cos^3(\theta)}, \\
z_3 &= \tan(\theta), \\
z_4 &= y - \frac{l}{2} \sin(\theta),
\end{align*}
\]

and

\[
\begin{align*}
u_1 &= \frac{v_1}{\rho \cos(\theta)}, \\
u_2 &= -\frac{3 \sin(\theta)}{l \cos^2(\theta)} \sin^2(\phi) v_1 + l \cos^3(\theta) \cos^2(\phi) v_2.
\end{align*}
\]

Under the transformations, kinematic model (3) can be mapped into the so-called chained form, that is,

\[
\begin{align*}
z_1 &= v_1, \\
z_2 &= v_2, \\
z_3 &= z_2 v_1, \\
z_4 &= z_3 v_1.
\end{align*}
\]

(6)

Remark 2.1: State and control transformations (4) and (5) are one-to-one if \( \theta \neq \pm \pi/2 \). This diffeomorphism can always be guaranteed by properly pre-rotating the \( x - y \) plane and making \( \theta(t) \in (-\pi/2, \pi/2) \) provided that boundary conditions \( \theta_0 \) and \( \theta_f \) on \( \theta \) satisfy the inequality \( |\theta_0 - \theta_f| < \pi \). If \( |\theta_0 - \theta_f| \geq \pi \), it becomes necessary to introduce an intermediate configuration with \( \theta_m \) such that \( |\theta_0 - \theta_m| < \pi \) and \( |\theta_m - \theta_f| < \pi \). In this case, trajectory planning from \( \theta_0 \) to \( \theta_f \) can be done by solving the problem twice, one from \( \theta_0 \) to \( \theta_m \) and another from \( \theta_m \) to \( \theta_f \).

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 (6) 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 (6). Then, given any boundary conditions $z(t_0) = z^0 = [z_{1,1}^0, z_{1,2}^0, z_{2,1}^0, z_{2,2}^0]^T$ and $z(T) = z^T = [z_{1,1}^T, z_{1,2}^T, z_{2,1}^T, z_{2,2}^T]^T$ (for some $T > 0$), 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 $z_{1,2}^T \neq z_{1,2}^T$ and that, through transformation (4), function $z_4 = F(z_1)$ also satisfies all the boundary conditions in the original state space.

**Remark 3.1:** If $z_{1,2}^T = z_{1,2}^T$, $v_1 = 0$ which causes a singularity in determining $v_2$. In this case, the singularity can be avoided by choosing an intermediate point $z_{3}^N$ with $z_{3}^N \neq z_{3}^N$ and by proceeding with planning two paths in the $z_1 - z_4$ plane.

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, it is assumed that $\phi_0 = \phi_f = 0$. Thus, for a feasible trajectory, the following boundary conditions on boundary points, slopes, and curvatures are applied: given $z_4 = F(z_1)$,

$$
\begin{align*}
    z_{1,1}^0 &= x_0 - \frac{1}{2} \cos(\theta_0), \\
    \frac{dz_4}{dz_1} \bigg|_{z_1 = z_{1,1}^0} &= \tan(\theta_0), \\
    \frac{d^2z_4}{dz_1^2} \bigg|_{z_1 = z_{1,1}^0} &= 0, \quad (7)
\end{align*}
\begin{align*}
    z_{1,2}^1 &= x_f - \frac{1}{2} \cos(\theta_f), \\
    \frac{dz_4}{dz_1} \bigg|_{z_1 = z_{1,2}^1} &= \tan(\theta_f), \\
    \frac{d^2z_4}{dz_1^2} \bigg|_{z_1 = z_{1,2}^1} &= 0. \quad (8)
\end{align*}
$$

**Remark 3.2:** If $\phi_0 = \phi_f = 0$ is not imposed, then the boundary curvatures in boundary conditions (7) and (8) should be changed to

$$
\left. \frac{d^2z_4}{dz_1^2} \right|_{z_1 = z_{1,1}^0} = \frac{\tan(\phi_0)}{l \cos^3(\theta_0)}, \quad \text{and} \quad \left. \frac{d^2z_4}{dz_1^2} \right|_{z_1 = z_{1,2}^1} = \frac{\tan(\phi_f)}{l \cos^3(\theta_f)}. \quad (9)
$$

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

B. Criterion for Avoiding Dynamic Objects

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

$$
v_{r,i} = v_{r} - v_{i}^k = \begin{bmatrix} v_{r,i,x}^k \\ v_{r,i,y}^k \end{bmatrix} = \begin{bmatrix} \dot{x} - v_{i,x}^k \\ \dot{y} - v_{i,y}^k \end{bmatrix}. \quad (10)
$$

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 $x_{i}' \in [x_{i}', x_{i}', t]$ with $x_{i}' = x_{i}^c - r_i - R$ and $x_{i}' = x_{i}^c + r_i + R$,

$$
(y_{i}' - y_{i}'^c)^2 + (x_{i}' - x_{i}'^c)^2 \geq (r_i + R)^2,
$$

where $\tau = t - (t_0 + kT_s)$ for $t \in [t_0 + kT_s, t_0 + T]$, $x_{i}' = x - v_{i,x}^c \tau$ and $y_{i}' = y - v_{i,y}^c \tau$ (which are the predictive velocities for the future). Note that the circle of radius $r_i + R$ for collision avoidance can be placed around the center of either the robot or the $i$th object. If the circle is placed around the robot, the collision avoidance criterion in the $y - x$ plane becomes: whenever $x_{i}' \in [x_{i}' - r_i - R, x_{i}' + r_i + R]$,

$$
(y_{i}' - y_{i}'^c)^2 + (x_{i}' - x_{i}'^c)^2 \geq (r_i + R)^2,
$$

where $\tau$, $x_{i}'$ and $y_{i}'$ are defined as before.

Fig. 4. Relative velocity of the robot with respect to the $i$th obstacle

It follows from state transformation (4) 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 \( z_i - z_1 \) space is: whenever \( x_i^k \in [z_i^1, 0.5l + 0.5l \cos(\phi) - r_i + R], \)
\[
\left( z_{4,i} + \frac{l}{2} \sin(\phi) - y_i^k \right)^2 + \left( z_{1,i} + \frac{l}{2} \cos(\phi) - x_i^k \right)^2 \geq (r_i + R)^2, \tag{12}
\]
where \( z_{1,i} = z_1 - v_{i,x} \tau \) and \( z_{4,i} = z_4 - v_{i,y} \tau \).

Note that, although \( \theta \) can be determined from \( z_3 \) and \( z_3 \) can be obtained as a result of applying lemma 1, exact mapping from \( z \) to \( (x, y, \theta) \) should not be used to numerically solve the problem of trajectory planning by imposing criterion (12). Instead, we choose to develop a new criterion only in terms of \( z_1 \) and \( z_4 \) (or \( z_{1,i}^k \) and \( z_{4,i}^k \)) so that analytical solution can be found for the problem of trajectory planning. To this end, note that all possible locations of point \( (x_i^k, y_i^k) \) are on the right semi circle centered at \( (z_{1,i}^k, z_{4,i}^k) \) 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 right semi circle renders the region from which the center of the \( i \)th object must stay clear, and the region is completely covered by the "unshaded" portion of the circle centered at \( (z_{1,i}^k, z_{4,i}^k) \) and of radius \( (r_i + R)/2 \).

Mathematically, the proposed collision avoidance criterion in the \( z_4 - z_1 \) plane is:
\[
(z_{1,i}^k - x_i^k)^2 + (z_{1,i}^k - y_i^k)^2 \geq (r_i + R + \frac{l}{2})^2, \tag{13}
\]
provided that
\[
x_i^k \in [z_{1,i}^k - r_i - R, z_{1,i}^k + 0.5l + r_i + R]. \tag{14}
\]

It is apparent from figure 5 that criterion (13) implies criterion (12) or (11). Once a steering method is chosen, the time interval during which criterion (13) should be imposed to avoid collision can be found from (14). That is, the proposed collision avoidance scheme has two parts: time criterion (14), and geometrical criterion (13).

C. A Feasible Collision-Free Trajectory Parameterization and Solution

A specific candidate class of feasible, collision-free trajectories are parameterized as
\[
z_4(z_1) = F(z_1) = a_k^T f(z_1), \tag{15}
\]
where \( a_k = [a_k^0, a_k^1, \ldots, a_k^n] \) is a constant vector to be determined, and \( f(z_1) = [1, z_1(t), (z_1(t))^2, \ldots, (z_1(t))^n]^T \) is the vector composed of basis functions of \( z_1(t) \). The following theorem is 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 (3) 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 \( q_0^k = [x_0, y_0, \theta_0, \phi_0]^T \) and \( q_f^k = [x_f, y_f, \theta_f, \phi_f]^T \) with \( \phi_0 = \phi_f = 0 \), as defined by (7) and (8), and satisfying the conditions that \( x_0 - \frac{l}{2} \sin(\theta_0) \geq x_f - \frac{l}{2} \sin(\theta_f) \) and that \( |\theta_0 - \theta_f| > \pi \), a collision-free path can be generated analytically by undertaking the following steps:

(i) Select coordinates \( (x, y) \) of the working space such that \( \theta \neq \pi/2 \), apply state and input transformations (4) and (5), determine the corresponding boundary conditions \( z_0^k = [z_{1,0}^k, z_{2,0}^k, z_{4,0}^k]^T \) and \( z_f^k = [z_{1,f}^k, z_{2,f}^k, z_{4,f}^k]^T \), and obtain the dynamics in chained form (6).

(ii) Let \( T \) be the time for the mobile robot to complete its maneuver and \( T_s \) be the sampling period such that \( k = T/T_s \) is an integer, that centers of objects \( O_i \) are 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 \( v_i^k \leq [v_{i,x}^k, v_{i,y}^k]^T \) for \( t \in [t_0 + kT_s, t_0 + (k+1)T_s] \). Then, for \( k = 0, \ldots, k-1 \), determine recursively constants \( a_k \) by ensuring the following second-order inequality (or inequalities): \( \forall i \in \{1, \ldots, n_o\} \)
\[
\min_{t \in [t_s^i, T_s^i]} g_2(z_1(t), k)(a_k^T a_k^T) + g_1.i(z_1(t), k, \tau)a_k^T + g_0.i(z_1(t), k, \tau)|\tau = t_0 = kT_s| \geq 0, \tag{16}
\]
where \( [t_s^i, T_s^i] \subset [t_0 + kT_s, T] \) is the time interval (if exists*) during which
\[
x_i^k \in [z_1(t) - v_{i,x}^k \tau - r_i - R, z_1(t) - v_{i,y}^k \tau + 0.5l + r_i + R], \tag{17}
\]
In (16), functions \( z_1(t), g_2(\cdot), g_1.i(\cdot) \) and \( g_0.i(\cdot) \) are defined as follows:
\[
z_1(t) = z_1^k + \frac{z_0^k - z_1^0}{T}(t - t_0 - kT_s), \ \forall t \in [t_0 + kT_s, T];
\]
\[
x_0^k = x_i(t_0), \ \ y_0^k = y_i(t_0);
\]
\[
x_i^k = t_0^k + T_s \sum_{j=0}^{k-1} v_{i,x}^j, \ \ y_i^k = y_0^k + T_s \sum_{j=0}^{k-1} v_{i,y}^j, \ \text{if } k > 0; \tag{18}
\]
*If the interval does not exist for some or all \( i \), inequality (16) is not needed for those objects.
(iii) A feasible, collision-free path of form (15) in the transformed state is found by solving $a^k$ according to

$$ a^k = \begin{bmatrix} a_0^k & a_1^k & a_2^k & a_3^k & a_4^k & a_5^k & a_6^k \end{bmatrix} = (B^k)^{-1}(Y^k - A^k a_0^k). $$

(iv) The steering inputs to achieve path (15) are given by, for $t \in (t_0 + kT_s, t_0 + (k + 1)T_s)$,

$$ v_1(t) = v_1 = \frac{z_1^f - z_0^i}{T_s}, $$

$$ v_2(t) = 6[a_3^k + 4a_4^k z_1^k + 10a_5^k (z_1^k)^2 + 20a_6^k (z_1^k)^3] v_1^r + 24[a_4^k + 5a_5^k z_1^k + 15a_6^k (z_1^k)^2] (t - t_0 - kT_s) v_1^r + 60[a_5^k + 6a_6^k z_1^k] (t - t_0 - kT_s)^2 v_1^r + 120a_6^k (t - t_0 - kT_s)^3 v_1^r. $$

(v) The corresponding feasible, collision-free Cartesian trajectory is given by $y = F(x - 0.5l \cos(\theta)) + 0.5l \sin(\theta)$, where $\theta$ can be found in closed form from state transformation (4) under steering inputs (22) and control mapping (5).

**Proof:** The proof, provided below according to the statements in the theorem, is done recursively for time intervals $t \in (t_0 + kT_s, t_0 + (k + 1)T_s)$, with boundary conditions in (7) and (8), and with intermediate boundary conditions in (18).

(i): Obvious from the discussions in section 2.

(ii) and (iii): Consider the class of candidate trajectories in (15). Applying boundary conditions either (7) or (18) and (8) results in equation (21). Note that the resulting matrix (or matrices) $B^k$ in (20) is (or are) nonsingular as long as $z_1^0 \neq z_1^f$. Hence, trajectories satisfying boundary conditions are parameterized in terms of $a_0^k$ as

$$ z_1 = \begin{bmatrix} (B^k)^{-1} (Y^k - A^k a_0^k) \end{bmatrix}^T f(z_1). $$

Substituting the above equation into (13) yields (16), a second order polynomial inequality in $a_0^k$.

(iv) To determine the steering control inputs, let $v_1(t) = C$ and $v_2(t) = C_0 + C_1 (t - t_0 - kT_s) + C_2 (t - t_0 - kT_s)^2 + C_3 (t - t_0 - kT_s)^3$ where $C$ and $C_j$ are constants. Directly integrating (6) yields

$$ z_1(t) = z_1^f + C (t - t_0 - kT_s), $$

$$ z_2(t) = z_2^f + C_0 (t - t_0 - kT_s) + \frac{C_1}{2} (t - t_0 - kT_s)^2 + \frac{C_2}{3} (t - t_0 - kT_s)^3 + \frac{C_3}{4} (t - t_0 - kT_s)^4, $$

$$ z_3(t) = z_3^f + C_2 (t - t_0 - kT_s) + \frac{CC_1}{2} (t - t_0 - kT_s)^2 + \frac{CC_2}{12} (t - t_0 - kT_s)^3 + \frac{CC_3}{20} (t - t_0 - kT_s)^5. $$

$z_4(t) = z_4^f + C_3 (t - t_0 - kT_s) + \frac{C^2 C_1}{2} (t - t_0 - kT_s)^3 + \frac{C^2 C_2}{24} (t - t_0 - kT_s)^4 + \frac{C^2 C_3}{120} (t - t_0 - kT_s)^5 + C_4 (t - t_0 - kT_s)^6.$

On the other hand, substituting $z_1(t) = z_1^f + C (t - t_0 - kT_s)$ into $z_2 = a^k f(z_1)$ yields

$$ z_4(t) = b_0 + b_1 (t - t_0 - kT_s) + b_2 (t - t_0 - kT_s)^2 + b_3 (t - t_0 - kT_s)^3 + b_4 (t - t_0 - kT_s)^4 + b_5 (t - t_0 - kT_s)^5 + b_6 (t - t_0 - kT_s)^6, $$

where $b_0 = \sum_{i=0}^{6} a_i^k (z_1^f)^i, b_1 = a_0^k C + a_1^k C + a_2^k C + a_3^k C + a_4^k C + a_5^k C + a_6^k C,$ $b_2 = a_2^k C^2 + a_3^k C^2 + a_4^k C^2 + a_5^k C^2 + a_6^k C^2,$ $b_3 = a_3^k C^3 + a_4^k C^3 + a_5^k C^3 + a_6^k C^3,$ $b_4 = a_4^k C^4 + a_5^k C^4 + a_6^k C^4,$ $b_5 = a_5^k C^5 + a_6^k C^5,$ and $b_6 = a_6^k C^5$. Then, in light of (21), we can solve for constants $C_i$ in $v_2(t)$ by comparing expressions (24) and (23). The result renders the steering inputs in (22).

(v): It is obvious from the discussions in the previous sections.
curvatures are nonzero (in a more general case), theorem 1 still holds except that $Y^k$ in (19) should be set to be

$$Y^k = \left[ z^k_1, z^k_2, z^k_3, y_f - \frac{l}{2} \sin(\theta_f), \tan(\theta_f), \frac{\tan(\phi_f)}{l \cos^3(\theta_f)} \right]^T,$$

where $\frac{\theta_f}{l} = \tan(\phi_f)/(l \cos^3(\theta_f))$.

**Remark 3.4:** If only constant velocity objects are present, one can simply choose $T = T_s$. In this case, we have $k = 1$ and $k = 0$; intermediate boundary conditions are no longer needed; and steering inputs in (22) are continuous. In particular, collision-free criterion (16) and the $z$-plane solution to trajectory planning become

$$g_2(z_1, k)(a_6)^2 + g_1(z_1, k; \tau)a_6 + g_0(z_1, k; \tau) \geq 0,$$

$$i = 1, \ldots, n_o \quad (25)$$

and

$$z_4 = f(z_1(t))(B^0)^{-1}(Y^o - A^0a_6) + a_6(z_1(t))^6, \quad (26)$$

where

$$g_2(z_1, k) = \left[ z^6_1 - f(z_1(t))(B^0)^{-1}A^o \right]^2.$$  

(27)

Nonetheless, even if all objects keep moving at constant speeds, choosing $k > 1$ may improve performance as it can approximate polynomials higher than 6th order.

**Remark 3.5:** It is apparent from steering inputs in (22) that, the larger the value of $T$ is chosen, the smaller the values of steering inputs become. As such, $T$ should be chosen to meet constraints on steering inputs and/or their rates of change. On the other hand, the collision avoidance capability will be physically curtailed if the values of steering inputs are limited, as will be shown in the next remark. To avoid dynamically moving objects, a robot must maneuver fast enough with respect to motions of the objects. In other words, as illustrated by the proposed criterion, the robot has to be able to properly adjust its relative velocities with respect to the moving objects.

**Remark 3.6:** Given the class of feasible trajectories in (15), collision avoidance in the presence of multiple moving objects depends upon solvability of inequality (16). Note that $g_2(z_1, k) \geq 0$. If $g_2(z_1, k) \neq 0$, inequalities in (16) belong to a family of parabolic functions with openings upward in the plane of $a_6^0$ versus $t$. Thus, as long as $g_2(z_1, k) \neq 0$, a solution to $a_6^0$ always exists no matter how many objects are considered. If $g_2(z_1, k) = 0$, $g_1(z_1, k; \tau) = 0$ as well, and little can be done to make inequality (16) valid.

To understand whether $g_2(z_1, k) = 0$ and its implications, consider the simpler case that $k = 1$. Similar discussions can be made for the case that $k > 1$. It follows from (26) that

$$z_4 = f(z_1(t))(B^0)^{-1}Y^o + a_6 \left[ z^6_1 - f(z_1(t))(B^0)^{-1}A^o \right].$$

Comparing with the expression of $g_2(z_1, k)$ in (27), we know that $g_2(z_1, k) = 0$ if, no matter what choice of $a_6$ is made, the feasible trajectory is degenerated into a quintic polynomial. Clearly, this is impossible for the function $z_4 = a f(z_1)$ except for the boundary points at which boundary conditions must hold for both 5th and 6th order polynomials and hence order degeneration occurs. At the boundary points, inequality (25) is not needed unless $t_0$ and/or $t_0 + T$ is in the interval $[t^*_1, t^*_2]$ as defined by (17). In the first case, collision has occurred already, and little can be done. In the second case, collision can be avoided by adjusting $T$ (unless one of the objects stays close enough to $z^f$).

In the vicinity of but not at the boundary conditions, $g_2(z_1, k)$ is close to being zero. If the interval (17) contains time instants corresponding to any of these points, one can form a proper solution to $a_6$ not by solving inequality (25), but by adjusting the robot speed and in turn the interval in (17) so it no longer contains any value very close to the boundary time instants. In summary, the theorem presents a solution to the trajectory planning problem in the presence of multiple dynamically moving objects.

**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 $q^T = (17, 10, -\frac{\pi}{4}, 0)$.
- **Moving obstacles:** $n_o = 3$, $O_1(t_0) = [5, 0]^T$, $O_2(t_0) = [9, 4]^T$, $O_3(t_0) = [19, 10]^T$ 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:**
  - $v^0_1 = [0, 0.3]^T$, $v^1_1 = [0.5, 0.2]^T$, $v^2_1 = [0.2, 0.2]^T$,
  - $v^0_2 = [-0.5, 0]^T$, $v^1_2 = [0.6, 0.1]^T$, $v^2_2 = [0.6, 0.1]^T$,
  - $v^0_3 = [-0.2, -0.4]^T$, $v^1_3 = [-0.2, 0.1]^T$, $v^2_3 = [-0.1, 0.1]^T$.
- **The solution to the parameterized trajectory:**
  - $a^0_6 = 2.9659 \times 10^{-5}$, $a^1_6 = 1.0577 \times 10^{-4}$, $a^2_6 = a^3_6 = 0.0013$.  

(28)

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

The simulation results are shown in figure 6. In figure 6, path 1 is the trajectory solution if the objects would maintain their velocities at $v^0_1$ for $t = [0, T]$, path 2 is the trajectory if the object velocities would be kept at $v^1_1$ for $t = [10, 40]$; and path 3 is the complete solution of the entire trajectory by considering all the speed changes of all objects. Positions of the robot and the three objects at $t = 0$ and every five seconds afterwards are marked by small circles along their corresponding trajectories, respectively. It is clear that, if path 1 is followed beyond $t = 10$, collisions between the robot and objects 1 and 3 will occur around $t = 30$ and $t = 35$ seconds, respectively. Similarly, if path 2 is followed beyond $t = 20$, a collision between the robot and object 3 will occur around $t = 35$ seconds. By design, path 3 is collision-free.
If the robot has a limited sensing range, piecewise solution to $a_k^0$ can be found by only considering the obstacles in the view. For example, if the sensor range is 10 (meters), solution $a_k^0$ changes from (28) to

$$a_k^0 = 2.9659 \times 10^{-5}, a_k^1 = 2.9659 \times 10^{-5}, a_k^2 = a_k^3 = 0.0016.$$  

In this case, the corresponding trajectories are plotted in figure 7, and path 3 is slightly changed.

![Fig. 7. Collision-free path of robot (solid line), obstacle 1 (dotted line), obstacle 2 (dashdot line) and obstacle 3 (dashed line)](image)

**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. Solvability condition is explicitly found. Effectiveness of the proposed method are illustrated by simulation results.

**REFERENCES**