



Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral

Tzu-Chen Liang^a, Jing-Sin Liu^{a,*}, Gau-Tin Hung^b, Yau-Zen Chang^b

^a *Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan 115, ROC*

^b *Department of Mechanical Engineering, Chang Gung University, Tao-Yuan, Taiwan, ROC*

Received 29 September 2003; received in revised form 18 May 2005; accepted 23 May 2005

Available online 27 June 2005

Abstract

This paper presents a nonholonomic path planning method, aiming at taking into considerations of curvature constraint, length minimization, and computational demand, for car-like mobile robot based on cubic spirals. The generated path is made up of at most five segments: at most two maximal-curvature cubic spiral segments with zero curvature at both ends in connection with up to three straight line segments. A numerically efficient process is presented to generate a Cartesian shortest path among the family of paths considered for a given pair of start and destination configurations. Our approach is resorted to minimization via linear programming over the sum of length of each path segment of paths synthesized based on minimal locomotion cubic spirals linking start and destination orientations through a selected intermediate orientation. The potential intermediate configurations are not necessarily selected from the symmetric mean circle for non-parallel start and destination orientations. The novelty of the presented path generation method based on cubic spirals is: (i) Practical: the implementation is straightforward so that the generation of feasible paths in an environment free of obstacles is efficient in a few milliseconds; (ii) Flexible: it lends itself to various generalizations: readily applicable to mobile robots capable of forward and backward motion and Dubins' car (i.e. car with only forward driving capability); well adapted to the incorporation of other constraints like wall-collision avoidance encountered in robot soccer games; straightforward extension to planning a path connecting an ordered sequence of target configurations in simple obstructed environment.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Car-like mobile robot; Trajectory generation; Shortest path; Nonholonomic

1. Introduction

Path planning problem [27] of autonomous mobile robots or vehicles (e.g. soccer robots), which the way mobile robots are able to change directions are restricted [20], has been widely studied in recent years to meet a variety

* Corresponding author. Tel.: +886 2 27883799; fax: +886 2 27824814.

E-mail address: liu@iis.sinica.edu.tw (J.-S. Liu).

of environmental constraints. The essential topic is to generate a set of paths to join two distinct configurations that meet certain path smoothness constraints. The common methodology for constructing feasible trajectories is by assembling arcs of simple curves [23]. Theoretical shortest path with bounded curvature synthesized by circular arcs of minimum radius tangentially connected by straight line segments was first presented to generate trajectories [8,9,23–25,30] for different environmental restrictions. A complete characterization of path synthesized by arcs of circle and straight line segments was addressed by [2], and shortest path by [9,30]. These studies concentrated on finding the path with theoretical minimal length from a family of simple curves. They showed such kind of paths has at most two cusps, where the robot changes its moving direction. However, its non-continuous curvature results in a control difficulty: at the junction of a straight line and an arc, mobile robot needs to stop its wheel motion to make the perfect tracking achievable [4]. For smoothing the discontinuity of the junction between line segment and circular arc, the clothoid was used as a transition curve to result in a continuous-curvature path [3,4,26]. Scheuer and Laugier [3] and Scheuer and Fraichard [4] incorporated a new constraint that the derivative of curvature is bounded into the path planning problem to make the planned path smoother. Different from Dubins' path, the curvature profile along the path has a trapezoid shape and is continuous. Kanayama and Miyake [10] proposed a pair of clothoid curves (or Cornu spirals), whose curvature varies linear with the arc length, to connect two straight lines to generate a smooth shortest path for a maximum jerk, which has continuous curvature.

Various primitives of curves were proposed to generate paths with continuous curvature for autonomous navigation of mobile robots or autonomous vehicles to guarantee good drive characteristics and small trajectory tracking errors in a wealth of work [10,12,15–17,20,21], e.g. B-spline [19], quintic polynomials for continuous curvature and velocity [16], quintic G^2 -spline [18]. Nelson [12] presented two types of paths, Cartesian quintics for lane changing maneuvers and polar splines for symmetric turns, which both can smoothly connect oriented straight line segments with zero curvature.

Polynomial spirals in general [22] (for example, clothoid and cubic spiral) are useful for trajectory generation because they provide a polynomial curvature profile that is easy to track, but are difficult to compute due to lack of closed-form expressions. In particular, a cubic spiral, whose curvature is a cubic function of arc length, is cut by [1] at its two inflection points to obtain a curve with zero curvature at both end-points. This portion of a cubic spiral can connect two configurations that are *symmetric*. A local path planning algorithm is presented for joining two configurations which are not symmetric by two cubic spirals through an intermediate configuration, which is called symmetric mean [1]. Path curvature and derivative of path curvature, or in physical terms the centripetal (lateral) acceleration and the variation of it, are used as cost function for path optimization intended to maximize passenger comfort.

However, some practical considerations, for example the bounded curvature constraint – a constraint in turning – of car-like mobile robots [13,20], and the avoidance of polygonal obstacles in obstructed environments [14,23,28,29], should be taken into consideration for a practical path planner. For moving inside a convex polygonal cell, a method of planning a path composed of minimum turning radius circular arcs and line segments was presented [24] where only a few boundary configurations have to be checked to avoid collision.

A path of low complexity [28], e.g. shorter path length and less number of reversals along the path, is preferred. A high value of curvature to make the generated path shorter may violate the curvature constraint happens in case the initial and end configurations are too close. Furthermore, the “smoothest” criterion makes the path overly long if two configurations are very distant. Despite the above complexity and geometric consideration a for path generation, also of main concern in the use of mobile robots is computational demand. In some applications like home care, there is acute computational need for the mobile robot to instantaneously in response to newly acquired environmental information. This demands a real-time generation of a feasible path on contemporary PCs, perhaps via an ordered sequence of destination configurations.

However, cubic spiral, which is parametrized by a few number of parameters, provides enough degrees of freedom to meet the curvature constraint. In this paper, we build on the work of [1] to present a practical and flexible path planning approach, which can be applied to the vehicle with or without backward motion capability in bounded plane, to remedy some drawbacks of cubic spiral path planning method [1]. A shortest path is generated from the

family of bounded-curvature paths consisting of at most two cubic spirals in connection with up to three straight line segments through a variation of potential intermediate configurations, notably not necessarily selected from the symmetric means circle.

The paper is organized as follows. In Section 2, the cubic spiral method is briefly reviewed, and the notations used in this paper are introduced. Section 3 is the synthesis of continuous and bounded curvature paths based on minimal locomotion cubic spirals linking start and destination orientation through a selected intermediate orientation. The numerical procedure to find a path of minimal length via change of potential intermediate orientations is presented in Section 4. Other characteristics regarding the generalizations of the path planning method to various situations will be presented in Section 5, including allowable motion direction constraint and wall collision avoidance. Some comparisons with [1] are also made. The last section is the conclusion.

2. Review of cubic spiral method

2.1. Notations and representation of a curve

In this paper, we follow most of notations of [1]. A triple $q \equiv (x, y, \theta)$ (or a posture of order one [20]) is to represent a vehicle configuration where (x, y) is the position and θ is the heading. A directed curve Π with finite length ℓ is defined by a triple:

$$\Pi \equiv (\ell, \kappa, q_0) \quad (1)$$

where $\kappa = d\theta/ds : [0, \ell] \rightarrow \mathbb{R}$ is the curvature and $q \equiv (x_0, y_0, \theta_0)$ is the initial configuration. The direction θ and position (x, y) at arc length s are defined by the integral equations describing the path starting from the initial configuration (x_0, y_0, θ_0) :

$$\theta(s) = \theta_0 + \int_0^s \kappa(t) dt, \quad x(s) = x_0 + \int_0^s \cos \theta(t) dt, \quad y(s) = y_0 + \int_0^s \sin \theta(t) dt \quad (2)$$

where s is defined as 0 at the initial point (x_0, y_0) . A configuration $q(s) = (x(s), y(s), \theta(s))$ is defined by this set of simultaneous Eq. (2).

Remark. Note that a posture (of order two) may be defined as more generally (x, y, θ, κ) [22].

2.2. Cubic spiral

By definition, cubic spiral is a set of trajectories that the direction function θ is a cubic polynomial of arc length. A portion of cubic spiral can be cut at its two inflection points whose curvature values are zero. The curvature function of this portion of cubic spiral with length ℓ is represented as a quadratic function of arc length s :

$$\kappa(s) = As(\ell - s)$$

where A is a nonzero constant to be determined. Its angle, which describes how much the curve turns from the initial orientation to final orientation, is denoted by

$$\alpha = \theta(\ell) - \theta(0) \quad (3)$$

From the first equation of (2) and the boundary conditions at $s=0$, and $s=\ell$, we have (Lemma 2, [1]):

$$\kappa(s) = \frac{6\alpha}{\ell^3} s(\ell - s) \quad (4)$$

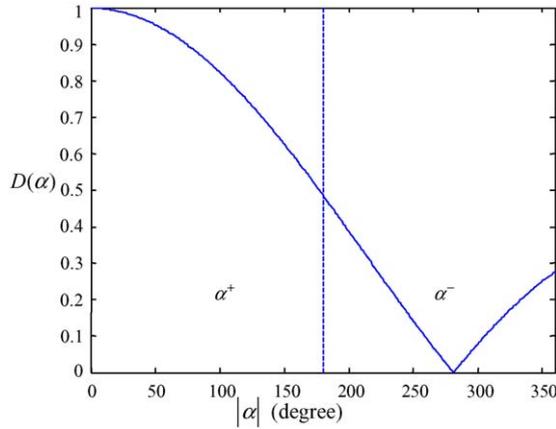


Fig. 1. Distance function of cubic spirals. The dashed vertical line corresponds to $|\alpha| = 180^\circ$; left side of this line is α^+ region and right side is α^- region [1].

If the length of a cubic spiral is 1, its size is given by (Lemma 3, [1]):

$$D(\alpha) \equiv 2 \int_0^{1/2} \cos \left(\alpha \left(\frac{3}{2} - 2t^2 \right) t \right) dt \tag{5}$$

This function can only be computed numerically, and the $D(\alpha)$ chart is replotted in Fig. 1. There is no closed form to represent the size of a cubic spiral of arbitrary length. Since all cubic spirals are similar, a pre-calculated $D(\alpha)$ table can evaluate the relation of ℓ and $d = \text{size}(q_1, q_2)$ by α using the following equation (Proposition 8, [1]):

$$\ell = \frac{d}{D(\alpha)} \tag{6}$$

2.3. Sketch of cubic spiral path planning method

2.3.1. Concept of symmetric configurations

For an arbitrary configuration q , $[q]$ denotes its position (x, y) , and (q) its direction θ . For a configuration pair (q_1, q_2) , the size is the distance between the two points $[q_1]$ and $[q_2]$, and the angle is the difference between the two directions (q_1) and (q_2) , i.e.:

$$\text{size}(q_1, q_2) \equiv d([q_1], [q_2]), \quad \text{angle}(q_1, q_2) \equiv \Phi((q_2) - (q_1)) \tag{7}$$

where the angle-normalizing function Φ is defined as

$$\Phi(\theta) \equiv \theta - 2\pi \left\lfloor \frac{\theta + \pi}{2\pi} \right\rfloor \tag{8}$$

If the function angle is applied to a vector \vec{v} , it means

$$\text{angle}(\vec{v}) = \text{atan2}(v_y, v_x) \tag{9}$$

v_x and v_y are scalars denote the x and y components of \vec{v} , respectively. If these two functions are applied to a cubic spiral, they are in fact applied to its two end-configurations.

A configuration pair $[q_1, q_2]$ is said to be parallel if $(q_1) = (q_2)$. A configuration pair $[q_1, q_2]$ is said to be symmetric if

$$\tan\left(\frac{\theta_1 + \theta_2}{2}\right) = \frac{y_2 - y_1}{x_2 - x_1}, \quad \text{if } x_1 \neq x_2 \quad \text{and} \quad \Phi\left(\frac{\theta_1 + \theta_2}{2}\right) = \pm \frac{\pi}{2}, \quad \text{if } x_1 = x_2, y_1 \neq y_2 \quad (10)$$

A *symmetric mean* q of any configuration pair (q_1, q_2) is a configuration that both (q_1, q) and (q, q_2) are symmetric pairs. All symmetric means of a configuration pair (q_1, q_2) forms a circle if $(q_1) \neq (q_2)$ or a line connecting (q_1, q_2) if $(q_1) = (q_2)$ (Proposition 3, [1]). The path planning algorithm of [1] is to choose one best symmetric mean q as an intermediate configuration of (q_1, q_2) so as to minimize the sum of cost functions of the cubic spiral joining the symmetric pair (q_1, q) and the cubic spiral joining the symmetric pair (q, q_2) . Two *smoothness* criteria based on minimization of the integration of centripetal force or the change of centripetal force are applied to optimize the selection.

It is noted that the symmetric property is essential in this method because a cubic spiral can connect two symmetric configurations.

2.3.2. Drawbacks

Kanayama and Hartman [1] proposed the use of cubic spirals, which is theoretically more meaningful than the set of clothoids, to generate the “smoothest” path for wheeled mobile robots. For example, smoother motion due to continuous curvature which is important for accurate tracking control, smaller maximal curvature which is especially important when a faster motion is needed. The characteristics of continuous curvature and criterion of minimal centripetal acceleration or minimal change of centripetal acceleration are indeed rational. There are, however, two main drawbacks not suitable for practical use in certain situations. The cost functions of Kanayama and Hartman [1] are for smoothness of paths based on either minimization of the integration of centripetal force or the change of centripetal force; the length of the path and maximal (or minimal) curvature along the path are not taken into consideration. As illustrated in Fig. 2, for configurations pairs with the same relative position and relative angle but varying size, the method generates similar “smoothest” path. However, the complexity of generated path is not satisfactory: the length of path is overly long when the size is large, and the maximal curvature is too high for turning when the size is small. Though Kanayama and Hartman [1] declared that this kind of configuration pair can be joined by simple curve (specifically, one symmetric curve), but for some cases, for example $q_1 = [0, 0, 0]$ and

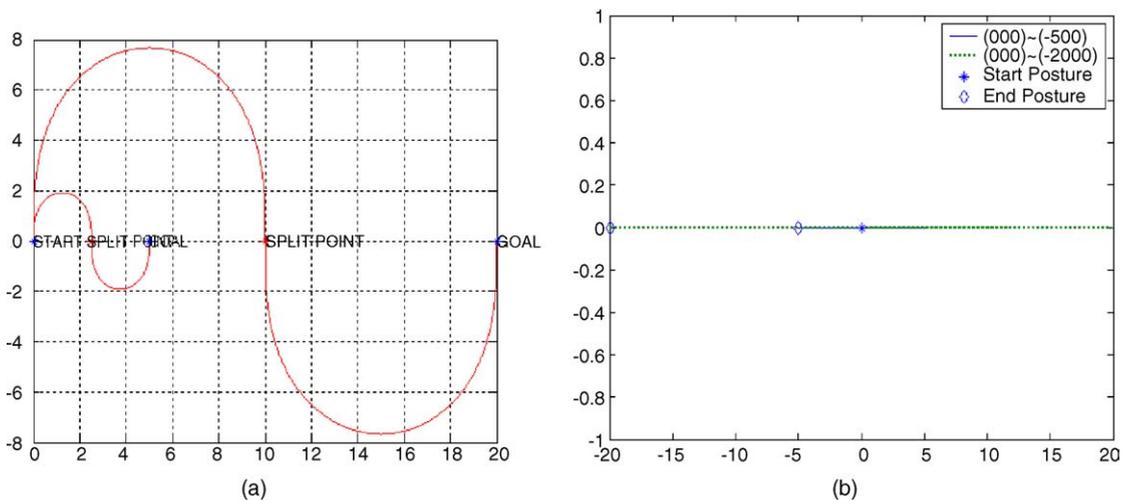


Fig. 2. Drawbacks of cubic spiral method based on symmetric mean: (a) curvature is large as the two locations are nearby, (b) infinite length path is generated.

$q_2 = [-a, 0, 0]$ configurations with the same horizontal heading but different positions, the simple curve may have infinite length (as figure shows). These motivate this study to remedy the drawbacks, while preserving the merits of the cubic spiral path planning method.

3. Generation of feasible paths using cubic spirals and line segments

The cubic spiral possesses nice properties suitable to be the primitive of trajectory generator of mobile robots. In particular, it is extensible by line segments: a path can be made up of connecting together cubic spirals and straight lines. Cubic spiral also has computational advantage in that it is parametrized by only a few number of parameters. In this paper, we consider the family of paths consisting of (at most) two cubic spirals and up to three line segments.

3.1. Constraint of maximal curvature

In practice, a wheeled mobile robot has its minimal radius of turning which is constrained by wheel arrangement [5–7]. This constraint may change dynamically according to driving velocity or control performance. Because curvature of a straight line is zero, the curvature constraint of the path is imposed on the cubic spiral segment. We use a constant κ_{\max} to describe the absolute value of maximal curvature of the planned path.

Consider a cubic spiral with an angle α (3). The curvature polynomial:

$$\kappa(s) = \frac{6\alpha}{\ell^3} s(\ell - s)$$

has the maximal (or minimal, if $\alpha < 0$) value at the middle point $s = \ell/2$:

$$\kappa_{\max} = \max(|\kappa(s)|) = \left| \kappa\left(\frac{\ell}{2}\right) \right| = \frac{3|\alpha|}{2\ell} = \frac{3|\alpha|D(\alpha)}{2d} \quad (11)$$

Hence, a constraint of curvature can be written as

$$|\kappa(s)| \leq \kappa_{\max}$$

It can be transformed to a size constraint of cubic spiral as

$$d \geq \frac{3|\alpha|D(\alpha)}{2\kappa_{\max}} \quad (12)$$

The minimal d can be solved as a function of α :

$$d_{\min}(\alpha) = \frac{3|\alpha|D(\alpha)}{2\kappa_{\max}} \quad (13)$$

3.2. Minimal locomotion by compound maximal-curvature cubic spirals

Consider a mobile robot that can move forward and backward. We will present the computation of two connecting maximal-curvature cubic spirals linking initial heading θ_1 through a given intermediate orientation θ_m to a desired orientation θ_2 from q_1 through a specified intermediate direction θ_m to achieve a goal direction given by (q_2), with the intermediate and end positions unspecified. This is called minimal locomotion by maximal-curvature cubic spiral starting at q_1 , which has the smallest size for given q_1, θ_m, θ_2 orientations. Since all cubic spirals are similar, longer curvature-constrained cubic spirals for transition from q_1 through θ_m to q_2 can be constructed from minimal locomotion

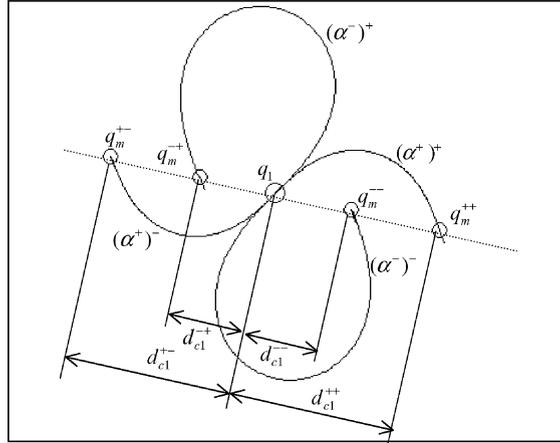


Fig. 3. There are four cubic spirals from an initial configuration q_1 to reach an intermediate direction (q_m) with maximal (or minimal) curvature value at their middle points.

Connecting an initial configuration q_1 which has direction θ_1 and an intermediate direction θ_m , there exist two cubic spirals with maximal curvature. Angles of these two cubic spirals are

$$\alpha_{c1}^+ = \Phi(\theta_m - \theta_1), \quad \alpha_{c1}^- = -\text{sgn}(\alpha_{c1}^+)(2\pi) + \alpha_{c1}^+ \quad (14)$$

where the angle of a cubic spiral is in the range $[-2\pi, 2\pi]$. If α_{c1}^+ is positive, then α_{c1}^- must be negative, and vice versa. The above notation (14) assures $\alpha_{c1}^+ \in [-\pi, \pi]$ and α_{c1}^- outside this range, as shown in Fig. 1. There are four cubic spirals with maximal curvature, denoted by

$$(\alpha_{c1}^+)^+, (\alpha_{c1}^+)^-, (\alpha_{c1}^-)^+, \text{ and } (\alpha_{c1}^-)^- \quad (15)$$

that achieve each of both angles (14), where positive/negative sign outside the parenthesis means forward/backward motion. An example of four cubic spirals with maximal curvature (15) is shown in Fig. 3. Clearly, their traveling distances are not the same due to different angles and motion directions.

For each of four cubic spirals (15), we can define:

- (i) four intermediate configurations q_m^{++} , q_m^{+-} , q_m^{-+} , and q_m^{--} , respectively. The first superscript denotes the range of the angle and the second superscript indicates forward or backward motion.
- (ii) four vectors:

$$\vec{v}_{c1}^{++} \equiv [q_m^{++}] - [q_1], \quad \vec{v}_{c1}^{+-} \equiv [q_m^{+-}] - [q_1], \quad \vec{v}_{c1}^{-+} \equiv [q_m^{-+}] - [q_1], \quad \vec{v}_{c1}^{--} \equiv [q_m^{--}] - [q_1] \quad (16)$$

- (iii) four distances for each path as the length of each vector in (16):

$$d_{c1}^{++} \equiv d_{\min}(\alpha_{c1}^+), \quad d_{c1}^{+-} \equiv -d_{\min}(\alpha_{c1}^+), \quad d_{c1}^{-+} \equiv d_{\min}(\alpha_{c1}^-), \quad d_{c1}^{--} \equiv -d_{\min}(\alpha_{c1}^-) \quad (17)$$

where $d_{\min}(\cdot)$ is always positive. Two of the above four d_{c1} are negative corresponding to backward motion.

A similar treatment holds for the angle $\alpha_{c2} = \Phi(\theta_2 - \theta_m)$: there are four paths connecting q_m and $(q_2) = \theta_2$ with corresponding vectors \vec{v}_{c2}^{++} , \vec{v}_{c2}^{+-} , \vec{v}_{c2}^{-+} , \vec{v}_{c2}^{--} , and length d_{c2}^{++} , d_{c2}^{+-} , d_{c2}^{-+} , d_{c2}^{--} .

As a result, for a given intermediate direction (θ_m), there are 16 different cubic spirals whose end direction is $(q_2) = \theta_2$ with maximal curvature. Only two of them lies on the symmetric means circle.

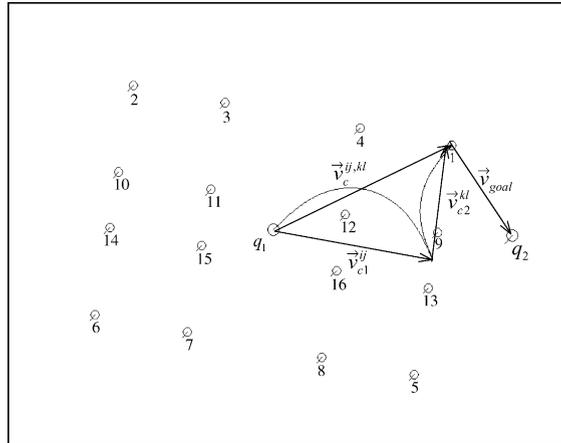


Fig. 4. Sixteen different end positions with the same orientation reachable from q_1 through minimal locomotions. One of the cubic spirals (i : +, j : +, k : +, l : -) is plotted to show the combination of minimal locomotion vectors.

Now consider two connecting cubic spirals linking initial heading θ_1 through a given intermediate orientation θ_m to a desired orientation θ_2 . Let the minimal locomotion vector of this composite cubic spiral be denoted by the vector $\vec{v}_c^{ij,kl}$. It is defined by the addition of minimal locomotion vector of each of the two cubic spiral segments:

$$\vec{v}_c^{ij,kl} = \vec{v}_{c1}^{ij} + \vec{v}_{c2}^{kl} = d_{c1}^{ij}\hat{n}_{c1} + d_{c2}^{kl}\hat{n}_{c2}, \quad \text{where the superscript } i, j, k, l \in \{+, -\} \tag{18}$$

where d_{c1}^{ij} , d_{c2}^{kl} are the corresponding minimum distances; \hat{n}_{c1} , \hat{n}_{c2} are the unit vectors representing forward motion direction. Fig. 4 shows the sixteen end locations of minimal locomotion of two connecting cubic spirals, where one combination of minimal locomotion vectors is plotted explicitly.

3.3. Synthesis of feasible paths

Following cubic spiral path planning method [1], we adopt two cubic spirals for left or right turn to form parts of a path. There are three zero curvature points at a two-cubic-spiral path. These three zero curvature points can be extended by up to three straight line segments to enhance reachability of path. Thus, to connect a given configuration pair at most five segments consisting of at most two cubic spirals and up to three straight line segments are used to synthesize a feasible path. The five directions of each path segment are denoted by

$$\theta_1, \theta_{c1}, \theta_m, \theta_{c2}, \theta_2 \tag{19}$$

where the angles of cubic spiral segments are

$$\theta_{c1} \equiv \Phi \left(\frac{\theta_1 + \theta_m}{2} \right), \quad \theta_{c2} \equiv \Phi \left(\frac{\theta_m + \theta_2}{2} \right) \tag{20}$$

with corresponding vectors:

$$\vec{v}_1 = d_1\hat{n}_1, \quad \vec{v}_{c1} = d_{c1}\hat{n}_{c1}, \quad \vec{v}_m = d_m\hat{n}_m, \quad \vec{v}_{c2} = d_{c2}\hat{n}_{c2}, \quad \vec{v}_2 = d_2\hat{n}_2 \tag{21}$$

where the positive (forward) directions of the five vectors denoted by the unit vectors \hat{n}_1 , \hat{n}_{c1} , \hat{n}_m , \hat{n}_{c2} , \hat{n}_2 are known, and the length of each vector d_1 , d_{c1} , d_m , d_{c2} , and d_2 remains undetermined. It is noted that positive (negative) value of each d denotes forward (backward) motion.

A feasible combination of these vectors satisfy the following synthesis equation:

$$[q_2] = [q_1] + \vec{v}_1 + \vec{v}_{c1} + \vec{v}_m + \vec{v}_{c2} + \vec{v}_2 \quad (22)$$

which can be rewritten more explicitly as

$$[q_2] = [q_1] + d_1 \vec{n}_1 + d_{c1} \vec{n}_{c1} + d_m \vec{n}_m + d_{c2} \vec{n}_{c2} + d_2 \vec{n}_2 \quad (23)$$

By definition (17), the constraints imposed on the two cubic spiral segments of a path are

$$\begin{cases} d_{c1} > d_{c1}^{ij} & \text{if } d_{c1}^{ij} > 0, \text{ else} \\ d_{c1} < d_{c1}^{ij} \\ d_{c2} > d_{c2}^{kl} & \text{if } d_{c2}^{kl} > 0, \text{ else} \\ d_{c2} < d_{c2}^{kl} \end{cases} \quad (24)$$

and the coefficients d_1 , d_m and d_2 , are free.

4. Shortest path synthesis based on cubic spiral primitive

The problem we address is to find a shortest path joining a given ordered pair of configurations (q_1, q_2) through a specified intermediate orientation. The family of paths for length minimization is the family of curves assembled by at most two cubic spirals and up to three straight-line segments. The path length is set as the cost function for optimization of paths [2,7–9,14], while maximal value of curvature is set as a constraint.

As the configuration pair is parallel, the shortest path is the line segment connecting them. For nonparallel configuration pair, to synthesize a shortest path through a specified intermediate orientation based on the cubic spirals, the first step is to synthesize a curvature-constrained path through a specified intermediate orientation, but without any length constraint imposed on the path generated. Then a minimization process is invoked to search a shortest path from the paths generated, as the minimal length criterion is imposed.

4.1. Imposing the criterion of minimal length

The objective of this subsection is to formulate a minimal length solution of the family of feasible paths with continuous and bounded curvature, composed of straight lines and cubic spirals, through a specified intermediate orientation. Assume θ_m and thus i, j, k , and l have been selected already. A cost function is associated with a candidate path from this family of paths. It is defined as the length of path. By summing up for the length of each segment, we obtain the expression for the length of a candidate path as

$$\text{cost}(\Pi) = |d_1| + \frac{|d_{c1}|}{D(\alpha_{c1}^i)} + |d_m| + \frac{|d_{c2}|}{D(\alpha_{c2}^k)} + |d_2| \quad (25)$$

where α_{c1}^i and α_{c2}^k are defined in (14).

For this problem, d_1, d_{c1}, d_m, d_{c2} , and d_2 should satisfy Eq. (23) subject to constraint (24) while minimizing cost function (25). Because d_{c1}^{ij} and d_{c2}^{kl} have been chosen, $D(\alpha_{c1}^i)$ and $D(\alpha_{c2}^k)$ are constants now. The cost of minimal locomotion is the constant:

$$\text{cost}(\Pi_{\theta_m}^{ij,kl}) = \frac{|d_{c1}^{ij}|}{D(\alpha_{c1}^i)} + \frac{|d_{c2}^{kl}|}{D(\alpha_{c2}^k)} \quad (26)$$

This part of cost is constant and cannot be further reduced. We define:

$$d_{c1}^* = d_{c1} - d_{c1}^{ij}, \quad d_{c2}^* = d_{c2} - d_{c2}^{kl} \quad (27)$$

Then to minimize (25) is equivalent to minimize the reduced cost:

$$\text{cost}^*(\Pi) = \text{cost}(\Pi) - \text{cost}(\Pi_{\theta_m}^{ij,kl}) = |d_1| + \frac{|d_{c1}^*|}{D(\alpha_{c1}^i)} + |d_m| + \frac{|d_{c2}^*|}{D(\alpha_{c2}^k)} + |d_2| \quad (28)$$

And constraints (24) can be rewritten as

$$d_{c1}^* > 0 \quad \text{if } d_{c1}^{ij} > 0, \quad \text{else } d_{c1}^* < 0, \quad d_{c2}^* > 0, \quad \text{if } d_{c2}^{kl} > 0, \quad \text{else } d_{c2}^* < 0 \quad (29)$$

4.2. Vectors that form a shortest path with given moving directions

Substituting (27) into (23), condition (23) for path assembling can be rewritten as

$$[q_2] = [q_1] + d_1 \hat{n}_1 + (d_{c1}^{ij} \hat{n}_{c1} + d_{c1}^* \hat{n}_{c1}) + d_m \hat{n}_m + (d_{c2}^{kl} \hat{n}_{c2} + d_{c2}^* \hat{n}_{c2}) + d_2 \hat{n}_2 \quad (30)$$

Define the vector:

$$\vec{v}_{\text{goal}} = [q_2] - [q_1] - \underbrace{(d_{c1}^{ij} \hat{n}_{c1} + d_{c2}^{kl} \hat{n}_{c2})}_{\vec{v}_c^{ij,kl}} \quad (31)$$

Then by (30), (31) can be written equivalently as

$$\vec{v}_{\text{goal}} = d_1 \hat{n}_1 + d_{c1}^* \hat{n}_{c1} + d_m \hat{n}_m + d_{c2}^* \hat{n}_{c2} + d_2 \hat{n}_2 \quad (32)$$

where the lengths d_1 , d_m and d_2 are free, and the signs of d_{c1}^* and d_{c2}^* are pre-decided by constraints (29). To solve the coefficients for a given \vec{v}_{goal} in (32), we define:

$$\hat{n}_{\Delta}^+ = \hat{n}_{\Delta} \quad \text{and} \quad \hat{n}_{\Delta}^- = -\hat{n}_{\Delta}, \quad \Delta \in \{1, c1, m, c2, 2\} \quad (33)$$

Define a set \mathbb{N} of unit vectors:

$$\mathbb{N} \equiv \{\hat{n}_1^+, \hat{n}_1^-, \hat{n}_m^+, \hat{n}_m^-, \hat{n}_2^+, \hat{n}_2^-, \hat{n}_{c1}^j, \hat{n}_{c2}^l\}, \quad j, l = +, - \quad (34)$$

and its corresponding set \mathbb{C} of coefficients which are either positive or zero:

$$\mathbb{C} \equiv \{d_1^+, d_1^-, d_m^+, d_m^-, d_2^+, d_2^-, d_{c1}^j, d_{c2}^l\} \quad (35)$$

where

$$d_{\Delta} = d_{\Delta}^+ - d_{\Delta}^-, \quad \Delta \in \{1, m, 2\}, \quad \text{and} \quad d_{c1}^j = |d_{c1}^*|, \quad d_{c2}^l = |d_{c2}^*| \quad (36)$$

Then (32) becomes:

$$\vec{v}_{\text{goal}} = d_1^+ \hat{n}_1^+ + d_1^- \hat{n}_1^- + d_{c1}^j \hat{n}_{c1}^j + d_m^+ \hat{n}_m^+ + d_m^- \hat{n}_m^- + d_{c2}^l \hat{n}_{c2}^l + d_2^+ \hat{n}_2^+ + d_2^- \hat{n}_2^- \quad (37)$$

By this transformation, all unknown coefficients in (37) are positive or zero, and we have total eight unknowns to be solved. The cost function (28) becomes:

$$\text{cost}^*(\Pi) = d_1^+ + d_1^- + \frac{d_{c1}^j}{D(\alpha_{c1}^i)} + d_m^+ + d_m^- + \frac{d_{c2}^l}{D(\alpha_{c2}^k)} + d_2^+ + d_2^- \quad (38)$$

Now the path length minimization can be formulated as a canonical form of linear programming problem:

$$\begin{aligned} & \text{maximize the linear function} && \text{cost}^*(\Pi) = c^T x \\ \text{(LP)} & \text{subject to the linear constraints} && Ax = b, \quad x \geq 0 \end{aligned}$$

where

$$c^T = \left[-1, -1, -1, -1, -1, -1, -\frac{1}{D(\alpha_{c1}^i)}, -\frac{1}{D(\alpha_{c2}^k)} \right]$$

$$x = [d_1^+, d_1^-, d_m^+, d_m^-, d_2^+, d_2^-, d_{c1}^j, d_{c2}^l]^T$$

$$A = \begin{bmatrix} \cos \theta_1 & -\cos \theta_1 & \cos \theta_m & -\cos \theta_m & \cos \theta_2 & -\cos \theta_2 & \text{sgn}(d_{c1}^j) \cdot \cos \theta_{c1} & \text{sgn}(d_{c2}^l) \cdot \cos \theta_{c2} \\ \sin \theta_1 & -\sin \theta_1 & \sin \theta_m & -\sin \theta_m & \sin \theta_2 & -\sin \theta_2 & \text{sgn}(d_{c1}^j) \cdot \sin \theta_{c1} & \text{sgn}(d_{c2}^l) \cdot \sin \theta_{c2} \end{bmatrix}$$

$$b = \vec{v}_{\text{goal}}$$

which can be solved by methods such as the simplex method [11] in a finite number of steps for the length of each path segment to satisfy the path length optimization objective as well as the curvature constraint. Because the optimal solution lies at one of the extreme points, it has only two non-zero elements, denoted as d_a and $d_b \in \mathbb{C}$. Assume d_a and $d_b \in \mathbb{C}$ are solved together with a corresponding pair of independent vectors \hat{n}_a and $\hat{n}_b \in \mathbb{N}$. Then the optimal solution to linear programming problem (LP) can be represented as

$$\vec{v}_{\text{goal}} = \vec{v}_a + \vec{v}_b = d_a \hat{n}_a + d_b \hat{n}_b \quad (39)$$

By Appendix A, d_a and d_b are given by

$$d_a = \|\vec{v}_{\text{goal}}\| \frac{-\sin \theta^-}{\sin(\theta^+ - \theta^-)}, \quad d_b = \|\vec{v}_{\text{goal}}\| \frac{\sin \theta^+}{\sin(\theta^+ - \theta^-)} \quad (40)$$

where

$$\theta^+ = \Phi(\text{angle}(\hat{n}_a) - \text{angle}(\vec{v}_{\text{goal}})), \quad \theta^- = \Phi(\text{angle}(\hat{n}_b) - \text{angle}(\vec{v}_{\text{goal}})) \quad (41)$$

Without loss of generality, \hat{n}_a and \hat{n}_b are defined such that θ^+ is positive and θ^- is negative. Then, by (39)–(41), the length of each path segment d_1 , d_{c1} , d_m , d_{c2} and d_2 in (23) are solved to minimize (25). A more efficient method based on some observations of the problem is presented in Appendix B.

4.3. Numerical procedure to find a shortest path

The procedure to find a shortest path joining two configurations is: search from all possible combination of (at most) two cubic spirals and up to three straight line segments, and find the one that minimizes the cost (25). Previous section has shown how to find the shortest path for a given intermediate orientation θ_m and given one of its 16 $\vec{v}_c^{ij,kl}$. To find the shortest path, using the procedure of previous section by varying θ_m from $-\pi$ to π and its corresponding 16 $\vec{v}_c^{ij,kl}$, the cost of each generated path is computed by the sum of generated five lengths d . The lowest cost path generated from this compute-and-compare procedure is the shortest path. The numerical procedure including collision-checking (discussed in Section 5.3) is summarized as follows: enumerating over all possible combinations of two cubic spirals and three straight line segments through variations of intermediate configurations,

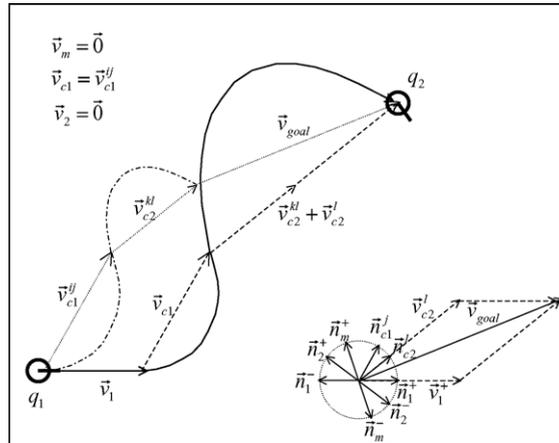


Fig. 5. This figure shows how to synthesize a shortest path if θ_m is given (zero in this case) and thus i, j, k, l have been decided. First the minimal locomotion vector $\vec{v}_{c1}^j + \vec{v}_{c2}^k$ is computed to define the \vec{v}_{goal} . Then two vectors to be extended are selected from eight candidates. In this case, \vec{v}_{c2}^l and \vec{v}_{c1}^+ are chosen. The resulting path (solid line) is composed of one straight line and two cubic spirals.

and finding the one that has shorter length and is collision free:

```

for  $\theta_m = -\pi$  to  $\pi$  step  $\Delta\theta$ 
  for  $i = +, -$  ( $\alpha_{c1}^i = \alpha_{c1}^-, \alpha_{c1}^+$ )
    for  $j = +, -$  (Direction  $C_1 =$  Forward, Backward)
      for  $k = +, -$  ( $\alpha_{c2}^k = \alpha_{c2}^-, \alpha_{c2}^+$ )
        for  $l = +, -$  (Direction  $C_2 =$  Forward, Backward)
          Check_Collision,
          if yes, discard the path
          if no, Compute_Cost(25)
        next
      next
    next
  next
next
next
next
next

```

(42)

where increment $\Delta\theta$ of numerical resolution can be chosen appropriately according to the requirements imposed.

Once a best θ_m for which the cost attains the minimum is found from the numerical process (42), the composite path of minimal length is generated. Fig. 5 shows an example to demonstrate the synthesis procedure of a shortest path via the synthesis of the vector \vec{v}_{goal} .

It is noted here that the selection of best θ_m direction is not necessarily selected from the symmetric means circle for non-parallel start and destination orientations (as [1] did), whose complete configuration (direction and orientation) is specified. Examples of synthesized shortest paths fully lying in the constrained plane are shown in Figs. 6 and 7.

4.4. A shortcut for search

If we set the minute increment $\Delta\theta = 0.0873$ in (42), for example, which is approximately 5° , then 1152 cases are searched to find a best one. In the case of a mobile robot that can move forward-and-backward in a free plane,

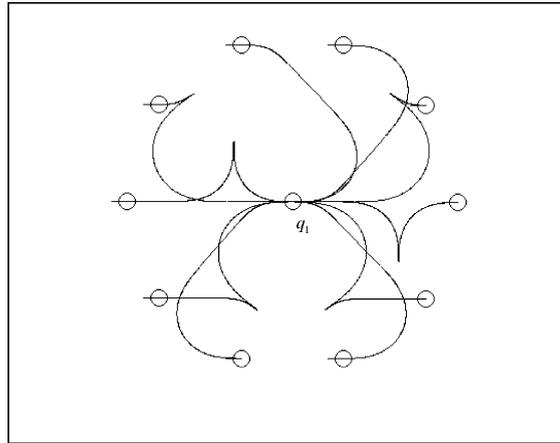


Fig. 6. An example of generated shortest paths. Each path has the same (q_2) but different $[q_2]$.

a cubic spiral with angle outside the range $[-\pi, \pi]$ seldom appears as a segment of shortest path, since intuitively backing up often increases the traveling length. The supporting reason is that: for a feasible path containing a cubic spiral with angle outside the range $[-\pi, \pi]$, then in most cases there exists at least one shorter path with angle inside the range $[-\pi, \pi]$. This observation can be employed to reduce the execution time by searching only the subset of paths with angle within $[-\pi, \pi]$. Following are the details of the argument.

Assume the shortest path contains one segment of cubic spiral with an angle α^- outside the range $[-\pi, \pi]$, then

$$d \geq d_{\min}(\alpha^-) \tag{43}$$

where d is the size of the cubic spiral. The length of this portion of path is

$$\ell_{\alpha^-} = \frac{d}{D(\alpha^-)} \tag{44}$$

On the other hand, there is another cubic spiral path with angle α^+ inside the range $[-\pi, \pi]$ also steers the vehicle to the same direction, without violating the maximal curvature constraint. Two cases need to be investigated:

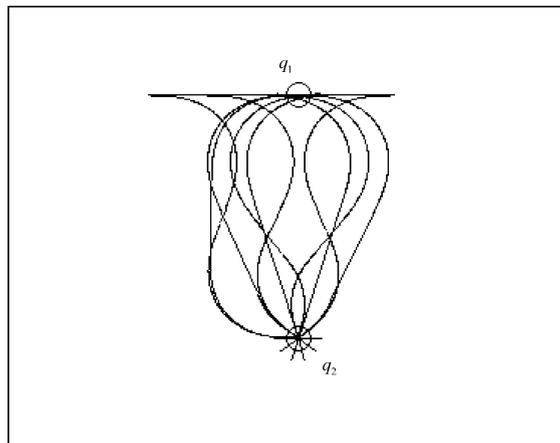


Fig. 7. Another example of generated shortest paths. Each path has the same $[q_2]$ but different (q_2) .

- Case 1: $d \geq d_{\min}(\alpha^+)$ In this case, an alternative path can be constructed by a single cubic spiral with angle α^+ and size d . The path length is

$$\ell_{\alpha^+} = \frac{d}{D(\alpha^+)} \tag{45}$$

By Fig. 1, for any two angles β_1 and β_2 with $|\beta_1| < \pi$ and $|\beta_2| < \pi$, we have

$$D(\beta_1) > D(\beta_2) > 0 \tag{46}$$

Thus,

$$\frac{d}{D(\alpha^+)} < \frac{d}{D(\alpha^-)}, \text{ i.e. } \ell_{\alpha^+} < \ell_{\alpha^-} \tag{47}$$

A shorter cubic spiral path with angle α^+ and size d is found to replace the one with angle α^- and size d .

- Case 2: $d \leq d_{\min}(\alpha^+)$

The alternative path can be constructed by a single cubic spiral with angle α^+ and size $d_{\min}(\alpha^+)$ and two connecting straight lines. The path length is computed as

$$\ell_{\alpha^+} = \frac{d_{\min}(\alpha^+)}{D(\alpha^+)} + (d_{\min}(\alpha^+) - d) \frac{1}{\cos\left(\frac{\alpha^+}{2}\right)} \tag{48}$$

Then $\ell_{\alpha^+} \geq \ell_{\alpha^-}$, if

$$\frac{d_{\min}(\alpha^+)}{D(\alpha^+)} + (d_{\min}(\alpha^+) - d) \frac{1}{\cos\left(\frac{\alpha^+}{2}\right)} - \frac{d}{D(\alpha^-)} \geq 0 \tag{49}$$

The above condition (49) can be rearranged compactly as

$$\frac{\frac{1}{D(\alpha^+)} + S(\alpha^+)}{\frac{1}{D(\alpha^-)} + S(\alpha^+)} \geq \frac{d}{d_{\min}(\alpha^+)} \tag{50}$$

Multiplying both side of (50) by the positive parameter $d_{\min}(\alpha^+)/d_{\min}(\alpha^-)$, we get

$$\frac{\frac{1}{D(\alpha^+)} + S(\alpha^+)}{\frac{1}{D(\alpha^-)} + S(\alpha^+)} \frac{d_{\min}(\alpha^+)}{d_{\min}(\alpha^-)} \geq \frac{d}{d_{\min}(\alpha^-)} \tag{51}$$

Fig. 10 plots the value of LHS of (51). By the assumption, RHS of (51) must be larger than or equal to 1, but from Fig. 10, when $|\alpha^+| < 139^\circ$, all values of LHS of (51) are less than 1. Furthermore, only when $(d/d_{\min}(\alpha^-)) < 1.0734$, (51) is possible to hold. The region where $\ell_{\alpha^+} \geq \ell_{\alpha^-}$ is shown in the gray area of Fig. 10. Apparently it is a relatively small region. Therefore, it is concluded that the case that cubic spiral with angle α^- cannot be replaced by one with corresponding α^+ seldom occurs.

This completes the argument behind the observation: a shorter path is very often found to replace the original cubic spiral with angle α^- (see Fig. 11). Thus, cubic spirals with angle α^- seldom appear as segments of a shortest-length path. This suggests a shortcut for path generation: for a mobile robot that can move forward and backward in free space, there is no need to check the α_{c1}^- and α_{c2}^- cases in path generation. This can reduce half of computational load with little length optimization sacrifice.

5. Other characteristics

Some other features of this path planning method are discussed in this section. Our shortest path planning method can be generalized to various situations: It is applicable to mobile robots without backward motion capability. It also has simple wall-collision avoidance abilities.

5.1. Dubin’s car (forward motion only): comparisons with [1]

The path planning algorithm is designed for the vehicle which can drive both forward and backward. However, with little and straightforward modification it also can be applied to nonholonomic mobile robots that can drive forward only (i.e. Dubin’s car). The modification is: let Direction C_1 and Direction C_2 be forward, and replace \mathbb{N} and \mathbb{C} by

$$\mathbb{N}_f \equiv \{\vec{n}_1^+, \vec{n}_m^+, \vec{n}_2^+, \vec{n}_{c1}^+, \vec{n}_{c2}^+\}, \quad \mathbb{C}_f \equiv \{d_1^+, d_m^+, d_2^+, d_{c1}^+, d_{c2}^+\} \tag{52}$$

In addition, when solving (39), the requirement for positive solution (of d_a and d_b) is

$$\theta^+ < \frac{\pi}{2} \text{ and } \theta^- > -\frac{\pi}{2} \tag{53}$$

For Dubins’ car (i.e. admitting only forward motion), a solution of (\vec{v}_a, \vec{v}_b) satisfying (39) might not exist for certain θ_m and $\vec{v}_c^{ij,kl}$, so that the algorithm is failed to generate a path. To avoid choosing such a case, its cost is set to be an extremely large value.

Figs. 8 and 9 show the planned paths for Dubin’s car for the same end configurations of Figs. 6 and 7, respectively. For comparisons, both our approach and [1] are shown in the same plot. It is clear that our approach generates a path of minimum radius of turning and very often is much shorter, while [1] generates a smoothest but much longer path and violates the curvature constraint. Detailed comparisons are tabulated in Tables 1 and 2.

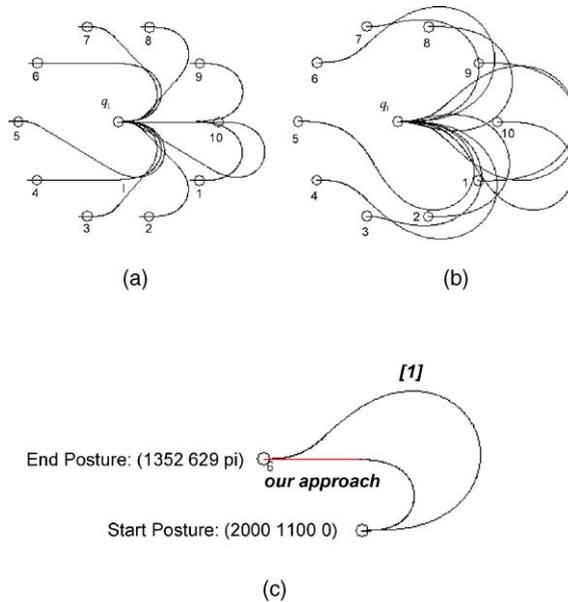


Fig. 8. Example of path planning for Dubin’s car for Fig. 6. Each path has the same goal direction (q_2) but different $[q_2]$: (a) our approach, (b) [1], and (c) a sampled path (path 6) of (a) and (b).

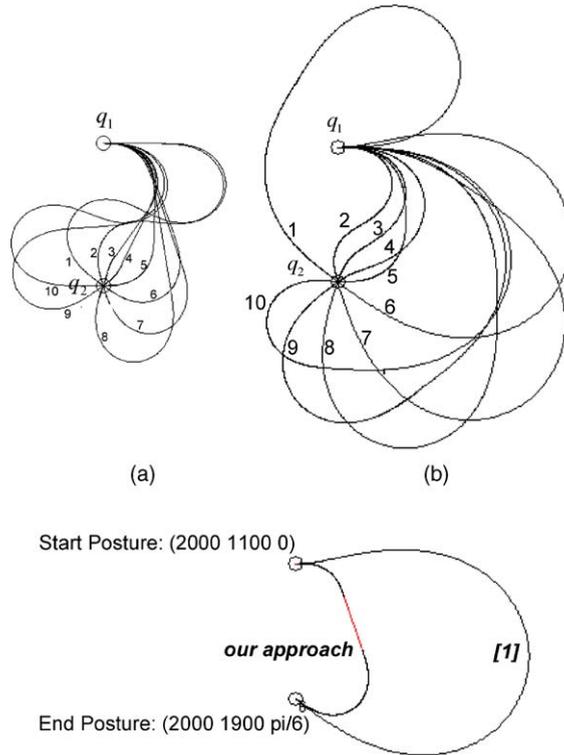


Fig. 9. Example of path planning for Dubin's car for Fig. 7. Each path has the same goal position [q_2] but different (q_2): (a) our approach, (b) [1], and (c) a sampled path (path 6) (a) and (b).

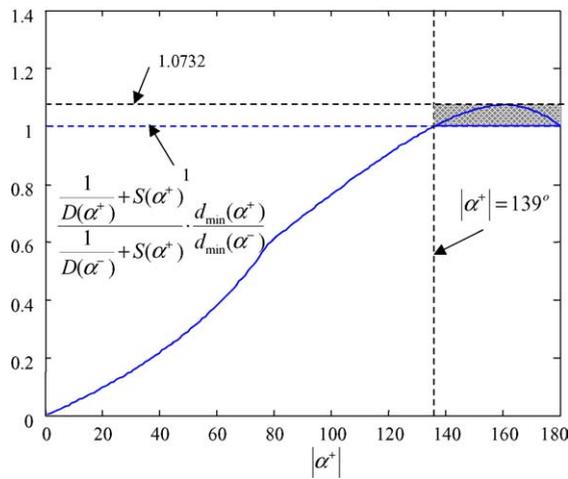


Fig. 10. The numerical value of LHS of (51).

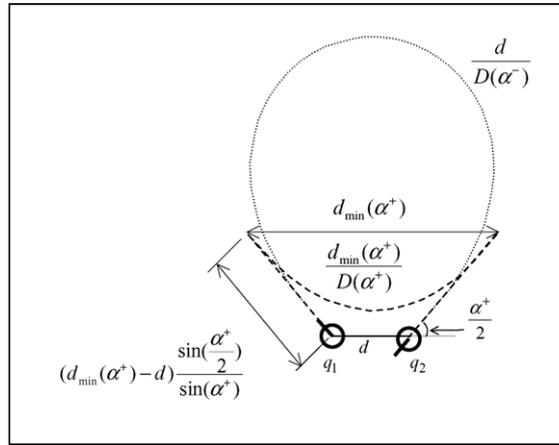


Fig. 11. This figure shows the generation of an alternative path by α^+ cubic spiral (dashed line) to replace an α^- one (dotted line) if $d < d_{\min}(\alpha^+)$. The shorter path is made up of two straight line segments and an α^+ cubic spiral.

Table 1
Compare the length and maximal curvature of the paths for Fig. 8

Path	Our approach			[1]	
	Length	No. of lines	Maximal curvature	Length	Maximal curvature
1	1581.604	2	0.005 (Given)	2835.669	2.4325E-03
2	1259.333	2		1996.644	2.7439E-03
3	1268.956	2		2054.655	2.6745E-03
4	1586.517	2		2902.866	2.5194E-03
5	1902.475	2		2710.05	2.0543E-03
6	1581.592	2		2835.241	2.4185E-03
7	1259.342	2		1996.43	1.9297E-03
8	1268.841	2		2054.41	2.6793E-03
9	1586.428	2		2902.403	2.3779E-03
10	1902.409	2		2709.564	3.1801E-03

Table 2
Compare the length and maximal curvature of the paths of Fig. 9

Path	Our approach			[1]	
	Length	No. of lines	Maximal curvature	Length	Maximal curvature
1	2305.699	2	0.005 (Given)	3746.283	1.5864E-03
2	1932.876	2		1171.6	5.0841E-03
3	1095.285	2		1240.915	3.7182E-03
4	1088.594	2		1436.491	3.1819E-03
5	1178.998	2		1324.303	3.5598E-03
6	1475.22	2		3534.677	1.7786E-03
7	1808.361	1		3988.471	1.7970E-03
8	2108.055	2		3968.48	2.1618E-03
9	2237.158	2		3995.483	2.8484E-03
10	2267.943	1		3891.327	4.4055E-03

It is noted that the length of second path solved by [1] is smaller than the path solved by our approach.

5.2. On wall-collision avoidance

In practice, there may exist regions where mobile robot is prohibited for passing to keep a distance away from human or machinery in the workcell. For example, in robot soccer game mobile robots move in a workspace bounded by wall-like boundaries and there are moving obstacles in the workspace. Our cubic spiral based path planning method is flexible to incorporate the space/positional constraints so that many acceptable wall collision-free paths for mobile robots can be efficiently generated. In what follows, we consider wall collision avoidance by introducing an additional term in the cost function.

A wall-type obstacle in a planar environment is described as a straight line segment equation

$$ax + by = c, (x, y) \in [x_l, x_u] \times [y_l, y_u] \tag{54}$$

In general, it is difficult to know whether and where a cubic spiral intersects with a line or not, since there are infinitely many points along the curve itself that needs to be checked. To achieve real-time performance for collision avoidance, we make some approximations in the following. There are six points on generated path that can be readily computed when \mathbb{C} is solved. The six points are

$$\begin{aligned} p_0 &= [q_1], & p_1 &= p_0 + \vec{v}_1, & p_2 &= p_1 + \vec{v}_{c1}, & p_3 &= p_2 + \vec{v}_m, & p_4 &= p_3 + \vec{v}_{c2}, \\ p_5 &= p_4 + \vec{v}_2 = [q_2] \end{aligned} \tag{55}$$

where $p_i \equiv (p_{ix}, p_{iy})$. Some of these points may be identical, because some d 's are zero.

Note that the initial and goal points (p_0 and p_5) are collision-free, so they must lie on the same side of wall. Thus, all we have to check are if

$$O(p_0) \cdot O(p_i) > 0, \quad \text{for } i = 1, 2, 3, 4 \tag{56}$$

where we define:

$$O(p) = ax + by - c \tag{57}$$

Then all six points defined in (55) are collision-free.

For checking whether a generated path, which is made up of straight line and cubic spiral segments, collides with the walls, we should check separately the collision of straight line segments and cubic spiral segments with the walls. For checking the collision of a straight line segment of a path with a wall (54), it suffices to check if two endpoints of straight line segment are on the same side of wall. On the other hand, for collision checking of a cubic spiral segment with a wall, we should compute the coordinate of selected points on a cubic spiral by generalized Fresnel integral equation (2), whose integration is a major computational burden [22]. Instead, we implement a simplified method, where only a few points are to be checked, to allow a simple runtime approximate collision checking, as described below.

Consider a cubic spiral with size d_{c1} , angle α and direction \vec{v} . As shown in Fig. 12, three straight line segments, whose total length equals the length of cubic spiral, are generated to approximately fit a cubic spiral. Length of the two end line segments is supposed the same and is denoted by l_{c1} , while the middle line segment has direction \vec{v} with length h_{c1} . Then

$$\frac{d_{c1}}{D(\alpha)} = h_{c1} + |d_{c1} - h_{c1}|S(\alpha), \quad S(\alpha) \equiv \frac{1}{\cos(\frac{\alpha}{2})} \tag{58}$$

Rearranging the above equation, we obtain

$$\frac{h_{c1}}{d_{c1}} = \frac{\frac{1}{D(\alpha)} - \text{sgn}(d_{c1} - h_{c1})S(\alpha)}{1 - \text{sgn}(d_{c1} - h_{c1})S(\alpha)} \tag{59}$$

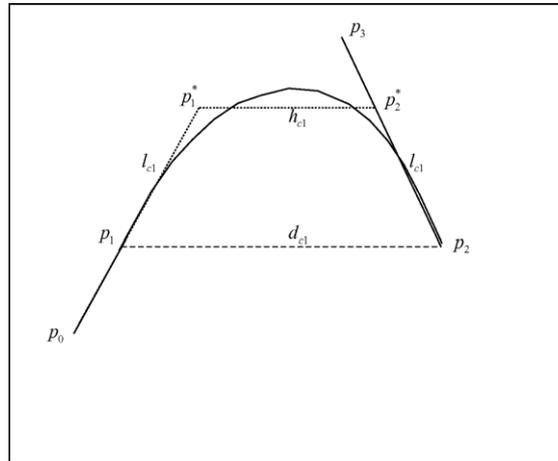


Fig. 12. This figure shows how to generate two more check points p_1^* and p_2^* from p_1 and p_2 . Note that p_1^* will replace p_1 in checking Eq. (46), but p_2^* will not replace p_2 . This is because p_2^* lies between p_2 and p_3 , and we need only check two endpoints of a straight line.

$$l_{c1} = \frac{|d_{c1} - h_{c1}|}{2 \cos\left(\frac{\alpha}{2}\right)} \quad (60)$$

From Fig. 12, the two more junctions of line segments are used as check points, whose coordinates are computed by

$$p_1^* = p_1 + l_{c1} \cdot \vec{n}_1, \quad p_2^* = p_2 - l_{c1} \cdot \vec{n}_m \quad (61)$$

These two points are close to the original cubic spiral, and check them can be a good approximation of collision detection of this segment. Checking the additional two points of (61) or original end points is dependent on the motion direction change of the path. Furthermore, note that this two more check points are collinear with the two straight line segments that connect with this cubic spiral. For a line segment it suffices to check its two end points to know whether the collision with a wall happens or not. Therefore, the number of check points for approximately checking if a cubic spiral colliding with a wall in (42) is five or less. This shortcut is effective for eliminating most unqualified paths, where some examples of generated collision-free paths fully lying in the constrained plane are demonstrated in Fig. 13. The average runtime is 0.3804 s.

5.3. Comparison with [1]

The family of paths used in [1] is continuous-curvature (“smoothest”) paths composed of two cubic spirals connecting two configurations through an intermediate configuration selected from a symmetric mean, which restricts the flexibility of path planner to incorporate other practical considerations like path length, curvature constraint, obstacle avoidance that we considered here. Instead, the family of paths we considered is made up of at most two curvature-constrained cubic spirals and up to three line segments connected via an intermediate orientation, which achieves a minimal length for shortest path among this family of curvature-constrained paths. The configuration reachability is enlarged. For example, the unattainable case of Fig. 2(b) by [1] mentioned in Section 2.3.2 is now solved in Fig. 14 by our approach.

The path planner of [1] is mainly for mobile robots with forward motion in free space, while our method is applicable to mobile robots with or without backward motion capability within wall-like boundaries. As compared to our method, Figs. 8 and 9 clearly show that longer paths are generated by the method [1], while the use of line

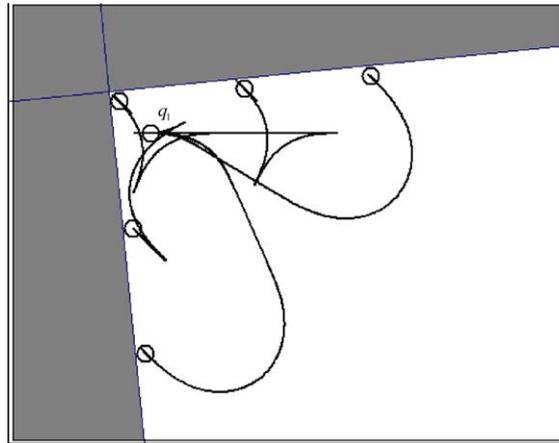


Fig. 13. Examples of the wall-collision avoidance path planning.

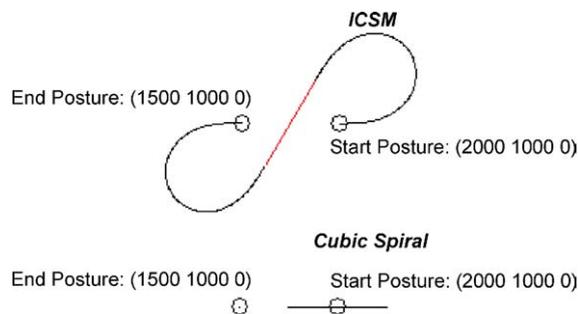


Fig. 14. Our approach generates a feasible path, while [1] cannot for the case of two end configurations with opposite headings of Fig. 2(b).

segments as part of the path is essential for synthesis of shorter paths. For computational efficiency, the runtime of both [1] and our approach is of the same order of magnitude, but [1] is a little faster in all simulations (0.0351 s versus 0.0461 s on average for Fig. 8, and 0.0340 s versus 0.0451 s on average for Fig. 9) that we have performed in (Centrino) PM-1.6 GHz, 512 MB laptop PC.

6. Conclusion

For generation of paths linking two configurations for wheeled mobile robots with turning radius lower bounded, this paper considers the family of curvature-constrained curves constituted by (at most) two cubic spirals to achieve reorientation, in connection with (up to) three straight line segments at inflection points of cubic spirals for length minimization. The reachability is enlarged due to the addition of line segments as parts of the cubic spiral path. For nonparallel configuration pair, a shortest path generation method searches a length-minimizing path from all feasible paths of the family generated by linear programming optimization over the length of each path segment via the selection of potential intermediate configurations, which are not necessarily selected from the symmetric mean. For efficiency enhancement, a faster search of feasible and smooth path is also presented at the expense of little

optimization loss. The easy-to-implement method lends itself to various generalizations. It is applicable to mobile robots with or without backward driving capability. Furthermore, it is flexible to incorporate simplified collision checking to rapidly generate wall-collision avoidance paths, at the expense of increasing the resulting path length, for autonomously navigating a mobile robot in a constrained plane, e.g. robot soccer game. As is clear, applying iteratively our approach also allows a straightforward extension to efficiently planning a feasible path connecting an ordered sequence of target configurations in simple obstructed environment. Simulation results are shown to demonstrate across a wide range of situations under various constraints. Extensive simulations with comparisons to existing planning methods will be conducted in the future work. Furthermore, we will apply our cubic spiral path planning method to dynamic environment where multiple objects are moving simultaneously, such as the situations encountered in robot soccer game [31].

Acknowledgments

This research was supported by National Science Council of ROC under contract NSC 91-2212-E-001-001. The authors thank the reviewers for helpful comments and the Co-Editor-in Chief (Asia) Prof. Tamio ARAI for assistance in preparing the revision.

Appendix A. Notes of $S(\alpha)$

It is often needed to represent a known vector \vec{v}_0 by two unit vectors \hat{n}_1 and \hat{n}_2 in our discussion. As shown in Fig. A1, let the angle between \hat{n}_1 and \vec{v}_0 is γ_1 ; \vec{v}_0 and \hat{n}_2 is γ_2 . Assume $\gamma_1 \neq \gamma_2$. Two lengths d_1 and d_2 can be solved from the identity:

$$\begin{bmatrix} v_{0x} \\ v_{0y} \end{bmatrix} = \begin{bmatrix} \cos \gamma_1 & \cos \gamma_2 \\ \sin \gamma_1 & \sin \gamma_2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

as

$$d_1 = \|\vec{v}_0\| \frac{-\sin(\gamma_2)}{\sin(\gamma_1 - \gamma_2)}, \quad d_2 = \|\vec{v}_0\| \frac{\sin(\gamma_1)}{\sin(\gamma_1 - \gamma_2)}$$

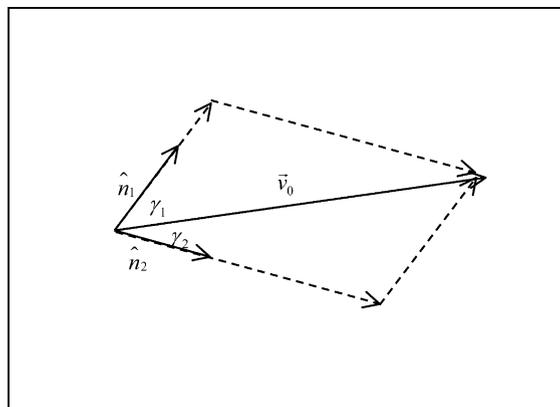


Fig. A1. Represent a known vector \vec{v}_0 by two independent vectors \vec{n}_1 and \vec{n}_2 . γ_1 and γ_2 are angles between (\vec{n}_1, \vec{v}_0) and (\vec{v}_0, \vec{n}_2) .

If the decomposition is symmetric, or $\gamma_2 = -\gamma_1$ we have the simple solution:

$$d_1 = d_2 = \|\vec{v}_0\| \frac{\sin(\gamma_1)}{\sin(2\gamma_1)} = \|\vec{v}_0\| \frac{1}{2 \cos(\gamma_1)}$$

and

$$d_1 + d_2 = \|\vec{v}_0\| \frac{2 \sin(\gamma_1)}{\sin(2\gamma_1)} = \|\vec{v}_0\| \frac{1}{\cos(\gamma_1)}$$

which is the form of $S(\alpha)$.

Appendix B. Faster method

This appendix presents a faster method to find which two constraints, i.e. which two rows of A , forms the extreme point.

Let

$$\Theta \equiv \{\text{Angle}(\text{angle}(\vec{n}) - \text{angle}(\vec{v}_{\text{goal}})) : \vec{n} \in \mathbb{N}\}$$

and define θ^{mp} is the minimal positive element of Θ , θ^{Mn} the maximal negative element of Θ , θ^{vmp} the vice minimal positive element of Θ , and θ^{vMn} is the vice maximal negative element of Θ .

There are three cases to be examined:

- (i) None of the corresponding \vec{n} 's of both θ^{mp} and θ^{Mn} is \vec{n}_{c1}^j or \vec{n}_{c2}^l , then

$$\theta^+ = \theta^{\text{mp}} \text{ and } \theta^- = \theta^{\text{Mn}}$$

- (ii) One of the corresponding \vec{n} of θ^{mp} or θ^{Mn} is \vec{n}_{c1}^j or \vec{n}_{c2}^l , then we should check θ^{vmp} or θ^{vMn} to compare which solution is maximal. Two possible combinations of solutions could exist in this case:

$$\theta^+ = \theta^{\text{vmp}} \text{ and } \theta^- = \theta^{\text{Mn}}, \quad \text{or} \quad \theta^+ = \theta^{\text{mp}} \text{ and } \theta^- = \theta^{\text{vMn}}$$

- (iii) The corresponding \vec{n} 's of θ^{mp} and θ^{Mn} are \vec{n}_{c1}^j and \vec{n}_{c2}^l , one more combination of solution:

$$\theta^+ = \theta^{\text{vmp}} \quad \text{and} \quad \theta^- = \theta^{\text{vMn}}$$

should be checked.

Therefore, in total at most four combinations of solutions are examined to find the extreme point.

References

- [1] Y.J. Kanayama, B.I. Hartman, Smooth local path planning for autonomous vehicles, *Int. J. Robot. Res.* 16 (3) (1997) 263–283.
- [2] P. Souères, J. Laumond, Shortest path synthesis for a car-like robot, *IEEE Trans. Automat. Contr.* 41 (5) (1996) 672–688.
- [3] A. Scheuer, Ch. Laugier, Planning sub-optimal and continuous-curvature paths for car-like robots, in: *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, BC, Canada, October 1998, 1998, pp. 25–31.
- [4] A. Scheuer, Th. Fraichard, Collision-free and continuous-curvature path planning for car-like robots, in: *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, 1997, pp. 867–873.
- [5] A.M. Shkel, V.J. Lumelsky, On calculation of optimal paths with constrained curvature: the case of long paths, in: *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996, 1996, pp. 3578–3583.
- [6] A.M. Shkel, V.J. Lumelsky, Curvature-constrained motion within a limited workspace, in: *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, 1997, pp. 1394–1399.

- [7] J. Boissonnat, A. Cérézo, J. Leblond, Shortest paths of bounded curvature in the plane, in: Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992, 1992, pp. 2315–2320.
- [8] L.E. Dubins, On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal position and tangents, *Am. J. Math.* 79 (1957) 497–516.
- [9] J.A. Reeds, R.A. Shepp, Optimal paths for a car that goes both forward and backwards, *Pac. J. Math.* 145 (2) (1990).
- [10] Y.J. Kanayama, N. Miyake, Trajectory generation for mobile robots *Robotic Research*, vol. 3, MIT Press, Cambridge, MA, 1986, pp. 333–340.
- [11] B. Kolman, R.E. Beck, *Elementary Linear Programming with Applications*, Academic Press, New York, 1980.
- [12] W. Nelson, Continuous-curvature paths for autonomous vehicles, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1989, pp. 1260–1264.
- [13] W.D. Esquivel, L.E. Chiang, Nonholonomic path planning among obstacles subject to curvature restrictions, *Robotica* 20 (2002) 49–58.
- [14] K. Jiang, L.D. Seneviratne, S.W.E. Earles, A shortest path based path planning algorithm for nonholonomic mobile robots, *J. Intell. Robot. Syst.* 24 (1999) 347–366.
- [15] R. Liscano, D. Green, Design and implementation of a trajectory generator for an indoor mobile robot, in: Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, 1989, pp. 380–385.
- [16] A. Takahashi, T. Hongo, Y. Ninomiya, G. Sugimoto, Local path planning and motion control for AGV in positioning, in: Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, 1989, pp. 392–397.
- [17] S. Fleury, P. Soueres, J.-P. Laumond, R. Chatila, Primitives for smoothing mobile robot trajectories, *IEEE Trans. Robot. Automat.* 11 (3) (1995) 441–448.
- [18] A. Piazzi, C. Guarino Lo Bianco, Quintic G^2 -splines for trajectory planning of autonomous vehicles, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2000, pp. 198–203.
- [19] K. Komoriya, K. Tanie, Trajectory design and control of a wheel-type mobile robot using B-spline curve, in: Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, 1989, pp. 398–405.
- [20] H. Delingette, M. Hebert, K. Ikeuchi, Trajectory generation with curvature constraint based on energy minimization, in: Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91, 1991, pp. 206–211.
- [21] P. Jacobs, J. Canny, Planning smooth paths for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, May 1989, pp. 2–7.
- [22] A. Kelly, B. Nagy, Reactive nonholonomic trajectory generation via parametric optimal control, *Int. J. Robot. Res.* 22 (7/8) (2003) 583–601.
- [23] F.G. Pin, H.A. Vasseur, Autonomous trajectory generation for mobile robots with non-holonomic and steering angle constraints, in: Proceedings of the IEEE International Workshop on Intelligent Motion Control, 1990, pp. 295–299.
- [24] H.A. Vasseur, F.G. Pin, J.R. Taylor, Navigation of a car-like mobile robot using a decomposition of the environment in convex cells, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1991, pp. 1496–1502.
- [25] G. Desaulniers, On shortest paths for a car-like robot maneuvering around obstacles, *Robot. Auton. Syst.* 17 (1996) 139–148.
- [26] Th. Fraichard, J.-M. Ahuactzin, Smooth path planning for cars, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2001, pp. 1722–1727.
- [27] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publisher, New York, 1991.
- [28] B. Mirtich, J. Canny, Using skeletons for nonholonomic path planning among obstacles, in: Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 1992, pp. 2533–2540.
- [29] B. Graf, J.M.H. Wadosell, C. Schaeffer, Flexible path planning for nonholonomic mobile robots, in: Proceedings of the Fourth European Workshop on Advanced Mobile Robots, 2001, pp. 199–206.
- [30] G. Desaulniers, F. Soumis, An efficient algorithm to find a shortest path for a car-like robot, *IEEE Trans. Robot. Autom.* 11 (6) (1995) 819–828.
- [31] J.-S. Liu, T.-C. Liang, Y.-A. Lin, Realization of a ball passing strategy in a robot soccer game: a case study of integrated planning and control, *Robotica* 22 (2004) 329–338.



Tzu-Chen Liang has been a Ph.D. candidate in aeronautics and astronautics, Stanford University since May, 2003. He received his bachelor and master degree in mechanical engineering, National Taiwan University in June, 1997 and June, 1999. He was employed as a research assistant in Institute of Information Science, Academia Sinica from June, 2001 to June, 2002. His primary research interests include topology optimization and algorithms in solving large numerical problems.



Jing-Sin Liu received the B.S. degree from the Department of Mechanical Engineering, National Cheng-Kung University in 1984, the M.S. degree in Mechanical Engineering from National Taiwan University in 1986, and the Ph.D. degree in Electrical Engineering from National Taiwan University in 1990. Since August 1990, he has been with Institute of Information Science, Academia Sinica, where he is currently an Associate Research Fellow. His current research interests include path planning, robot soccer game, nonlinear control theory and applications, and computer animation.



Gau-Tin Hung received the B.S. and the M.S. degrees from the Department of Mechanical Engineering, Chang Gung University, Tao-Yuan, Taiwan, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree in the same university. His current research interests include computer vision, real-time visual tracking, and soft-computing.



Yau-Zen Chang received the M.S. degree in Mechanical Engineering from National Taiwan University, Taipei, in 1986, and the D.Phil. degree in Information Engineering from the Department of Engineering Science, Oxford University, UK, in 1992. Since August 1993, he has been on the faculty of the Department of Mechanical Engineering at Chang Gung University and is currently an Associate Professor. His current research interests include genetic algorithms, artificial neural network, machine vision, and intelligent control systems. He is a member of IEEE.