

A Path Planning Method Using Cubic Spiral with Curvature Constraint

Tzu-Chen Liang and Jing-Sin Liu

Institute of Information Science 20, Academia Sinica, Nankang, Taipei 115, Taiwan, R.O.C.,

Email: hartree@iis.sinica.edu.tw and liu@iis.sinica.edu.tw

Abstract. This paper addresses a new path planning method, whose objective is to build a process to generate a path which connects two configurations of car-like mobile robots. The generated path is constituted by both cubic spirals and straight lines, and has continuous and bounded curvature. We will show the procedure to find the path with theoretical minimal length and to simplify it for the reason of practical use. Mobile robots with forward and backward driving abilities and only uni-direction driving ability are both considered. This method is flexible and is eligible to incorporate with other constraints like wall-collision avoidance. This will also be discussed.

I. Introduction

Path planning problem of autonomous mobile robot has been widely studied in recent years. The essential topic is to generate an acceptable path to join two distinct configurations with some constraints and optimize it. L. E. Dubins first discussed the shortest paths with bounded curvature synthesized by arc and straight line in 1957[8]. J. A. Reeds and L. A. Shepp further studied the same path with cusp, i.e. the vehicle can drive both forward and backward, in 1990[9]. A complete characterization of path synthesized by arc of circle and straight line was addressed by P. Souères and J. Laumond[2]. These studies concentrated on the finding of the path with theoretical minimal length. They showed such kind of path has at most two cusps. 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.

A. Scheuer and Ch. Laugier [3][4] added a new constraint that the derivative of

curvature is bounded to the path planning problem and made the planned path smoother. Different from Dubins' path, the curvature value along the path has a trapezoid shape and is continuous.

Y. Kanayama and N. Miyake suggested another set of curve called clothoid curves or Cornu spirals to form a smooth path which has continuous curvature [10]; A. Scheuer and Ch. Laugier in [4] also have a similar discussion. A more natural and smoother set of curve called cubic spiral is introduced by Kanayama and Hartman in 1997[1]. Cubic spiral is a kind of trajectory which direction function is cubic. Kanayama cut a cubic spiral in its two inflection points to obtain a curve with finite length and has zero curvature on both end-points. This portion of a cubic spiral can connect two configurations that are *symmetric*. Two configurations which are not symmetric can be joined through an intermediate configuration, which is called *symmetric mean*, by two cubic spirals. Instead of path length, Kanayama used path curvature and derivative of path curvature as criteria to optimize it. The physical meanings of these two criteria are in fact the centripetal (lateral) acceleration and the variation of it.

Though the cubic spiral method can generate smoother path than other ones, it does not consider the bounded curvature constraint of car-like mobile robots. An unreasonable high value of curvature happens when the initial and end configurations are too close, and the "smoothest" criteria make the path too long if two configurations are relative far.

In this paper, we address a new path planning method. Following Kanayama's cubic spiral method, we incorporate it with straight line segments in its zero curvature points. The result path has both continuous and bounded curvature. It has two cubic spiral segments and at most two straight line segments and is optimized by a shortest-length criterion. This method can be also applied to the vehicle with both forward and backward moving abilities.

In the next section, the cubic spiral method is briefly reviewed, and notations used in this paper are introduced. Section three we will address the improved cubic spiral method and the procedure to find a shortest path. Other characteristics will be discussed in section four, including wall collision avoidance and motion direction constraint of mobile robot. The last section is the conclusion.

II. Review of Cubic Spiral Method

II.1 Background

Y. J. Kanayama and B. I. Hartman suggested the using of cubic spirals to find the "smoothest" path for mobile robot path planning problem. They claimed the set of cubic spiral is theoretical more meaningful than the set of clothoids. The characteristics of continuous curvature and criterion of minimal centripetal

acceleration or minimal change of centripetal acceleration are indeed rational. However, this method has some constriction and drawbacks and is not suitable for practical use. Later we will briefly describe this method and its main drawbacks.

II.2 Notations and Representation of a Curve

In this paper, we follow most of Kanayama's notations. A triple $q \equiv (x, y, \theta)$ is to represent a vehicle configuration. 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.

$$\begin{aligned} size(q_1, q_2) &\equiv d([q_1], [q_2]) \\ angle(q_1, q_2) &\equiv \Phi((q_2) - (q_1)) \end{aligned} \quad (1)$$

where the angle-normalizing function Φ is defined as

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

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

$$angle(\vec{v}) = \text{atan2}(v_y, v_x) \quad (3)$$

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

A directed curve Π with finite length ℓ is defined by a triple,

$$\Pi \equiv (\ell, \kappa, q_0) \quad (4)$$

where $\kappa: [0, \ell] \rightarrow \mathbb{R}$ is its curvature and $q_0 \equiv (x_0, y_0, \theta_0)$ is its initial configuration (Lipshutz 1969). Its direction θ and position (x, y) at arc length s are evaluated by

$$\begin{aligned} \theta(s) &= \theta_0 + \int_0^s \kappa(t) dt \\ x(s) &= x_0 + \int_0^s \cos \theta(t) dt \\ y(s) &= y_0 + \int_0^s \sin \theta(t) dt \end{aligned} \quad (5)$$

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

II.3 Cubic Spirals

By definition, cubic spiral is a set of trajectories that their direction functions θ are cubic. An entire cubic spiral has infinite length, but the useful portion is cut from its two inflection points and has finite length. The curvature function of this portion of

cubic spiral with length ℓ is represented as

$$\kappa(s) = As(\ell - s) \quad (6)$$

where A is a constant to be determined. At the inflection points ($s = 0$ and $s = \ell$) the cubic spiral has zero curvature. The constant A of a cubic spiral joining two separated configurations which have relative angle of $\alpha = \theta(\ell) - \theta(0)$ can be solved by the first equation of (5) and the curvature function becomes,

$$\kappa(s) = \frac{6\alpha}{\ell^3}s(\ell - s) \quad (7)$$

If the length of a cubic spiral is 1, its *size* is given by[1]

$$D(\alpha) \equiv 2 \int_0^{1/2} \cos(\alpha(\frac{3}{2} - 2t^2)t) dt \quad (8)$$

There is no close form to represent the *size* of a cubic in any 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,

$$\ell = \frac{d}{D(\alpha)} \quad (9)$$

The $D(\alpha)$ table is plotted in Fig. 1.

II.4 Concept of Symmetric Configurations

A configuration pair $[q_1, q_2]$ is said to be symmetric if

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

The symmetric property is essential in this method because a cubic spiral can connect two symmetric configurations.

II.5 Sketch of Cubic Spiral Path Planning Method

Because one cubic spiral can connect two symmetric configurations, we need at least two cubic spirals to connect any two configurations. 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. Kanayama proved that all symmetric means of a configuration pair $[q_1, q_2]$ forms a circle. The optimization is to choose one symmetric mean q from this circle to minimize the cost functions.

II.6 Drawbacks

Kanayama's method can generate smoother path than other method, and is easy to be implemented by software programming. However, there are two main drawbacks of

this method that make it not fit practical use. First, the cost functions of Kanayama’s method either minimize the integration of centripetal force or the change of centripetal force; they do not consider the arc length and maximal (or minimal) curvature. As shown in Fig. 2, Configurations pairs which have the same relative position and relative *angle* but different *size* have similar “smoothest” path, yet the length of this path is too long when the size is large and the curvature is too high when the size is small.

Secondly, the method fails in the configuration pair that two configurations are originally symmetric. Though he declared this kind of configuration pair can be joined by simple curve (one symmetric curve), but in some cases, like $q_1 = [0,0,0]$ and $q_2 = [-a,0,0]$, the simple curve may have infinite length.

III. Our Solution: the Improved Cubic Spiral Method

The cubic spiral indeed has many good properties that are suitable to be the path of mobile robots. Therefore we improve Kanayama’s method to assemble a path by at most two cubic spirals and two straight lines. The assembled path still has continuous curvature. Different from original method, the improvement adds a maximal (or minimal) value of curvature as constraints and path length as cost criterion. It can be applied to path planning problem that allows or doesn’t allow backward motion. It also has simple wall-collision avoidance abilities. We will show the procedure to find the theoretical shortest path and some simplification to fit practical use.

III.1 Problem Statement

The problem is to find the path joining a given ordered pair of configurations $[q_1, q_2]$ with bounded and continuous curvature along it. The path is assembled by at most two cubic spirals and two straight lines and is optimized to be shortest.

III.2 Constraints of Maximal Curvature

In general, the path of a mobile robot has its minimal radius of curvature which is constrained by wheel arrangement[5][6][7]. This constraint also may dynamically changes according to driving velocity or control performance. We use a constant κ_{\max} to describe the absolute value of maximal curvature of the planned path. Because curvature of a straight line is zero, this constraint only affects the cubic spiral segment of the path.

For a specific cubic spiral with *angle* α , since $\kappa(s) = \frac{6\alpha}{\ell^3} s(\ell - s)$ is the curvature polynomial, it has the maximal (or minimal, if $\alpha < 0$) value at the middle point,

where $s = \frac{\ell}{2}$, and

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

Therefore if we have a constraint of curvature which is $|\kappa(s)| \leq \kappa_{\max}$. It can be transformed to the *size* constraint,

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

The minimal d can be solved as a function of α

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

III.2 Minimal Locomotion

In this section, we will show the computation of the minimal locomotion from q_1 through an intermediate direction θ_m to (q_2) . The maximal curvature constraint discussed in above section is applied, and there are 16 kinds of different minimal locomotion vector can be generated for a given θ_m .

We restrict the *angle* of a cubic spiral in the range $[-2\pi, 2\pi]$, for an initial configuration q_1 which has direction θ_1 to the intermediate configuration with direction θ_m , there exists two cubic spirals that all have the maximal curvature. *Angles* of these two cubic spirals are,

$$\begin{aligned} \alpha_{c1}^+ &= \Phi(\theta_m - \theta_1) \\ \alpha_{c1}^- &= -\text{sgn}(\alpha_{c1}^+) \cdot (2\pi) + \alpha_{c1}^+ \end{aligned} \quad (14)$$

If α_{c1}^+ is positive, then α_{c1}^- must be negative, and vice versa. The above notation assures that $\alpha_{c1}^+ \in [-\pi, \pi]$ and α_{c1}^- outside this range, as shown in Fig. 1. To achieve both angles there are two kinds of motion: forward and backward. Therefore the amounts of cubic spirals are four. We use another positive or negative sign to represent these four kinds of motion,

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

Positive sign outside the bracket means forward motion, and negative one means backward motion. Fig. 3 shows the four trajectories. Although they have the same κ_{\max} constraints, four traveling distance are different due to the angles and motion directions are different. These four intermediate configurations are named

$q_m^{++}, q_m^{+-}, q_m^{-+},$ and q_m^{--} respectively. The first superscript denotes the range of the angle and the second superscript represents the motion direction. The four distance are also named by the same notation,

$$\begin{aligned}
d_{c1}^{++} &\equiv d_{\min}(\alpha_{c1}^+) \\
d_{c1}^{+-} &\equiv -d_{\min}(\alpha_{c1}^+) \\
d_{c1}^{-+} &\equiv d_{\min}(\alpha_{c1}^-) \\
d_{c1}^{--} &\equiv -d_{\min}(\alpha_{c1}^-)
\end{aligned} \tag{16}$$

Note the result of $d_{\min}(\bullet)$ is always positive, but above four d_{c1} may be negative. The negative value means the motion is accomplished by backward motion of vehicle. Four vectors are defined as

$$\begin{aligned}
\vec{v}_{c1}^{++} &\equiv [q_m^{++}] - [q_1] \\
\vec{v}_{c1}^{+-} &\equiv [q_m^{+-}] - [q_1] \\
\vec{v}_{c1}^{-+} &\equiv [q_m^{-+}] - [q_1] \\
\vec{v}_{c1}^{--} &\equiv [q_m^{--}] - [q_1]
\end{aligned} \tag{17}$$

Use the same procedure again, if $\alpha_{c2} = \Phi(\theta_2 - \theta_m)$, there are also four trajectories exist for each q_m . The corresponded vectors are $\vec{v}_{c2}^{++}, \vec{v}_{c2}^{+-}, \vec{v}_{c2}^{-+}, \vec{v}_{c2}^{--}$, and their length are $d_{c2}^{++}, d_{c2}^{+-}, d_{c2}^{-+}, d_{c2}^{--}$. As a result, for a specific intermediate direction θ_m , there are 16 different paths with maximal curvature to arrive at the end direction θ_2 . The symmetric mean circle can only represent two of them.

The minimal locomotion vector of a given θ_m is $\vec{v}_c^{ij,kl}$, defined by

$$\begin{aligned}
\vec{v}_c^{ij,kl} &= \vec{v}_{c1}^{ij} + \vec{v}_{c2}^{kl} \quad \text{where,} \\
i, j, k, l &\in \{+, -\}
\end{aligned} \tag{18}$$

corresponded minimal distance can be also represented as d_{c1}^{ij} and d_{c2}^{kl} . Fig. 4 shows the sixteen end positions of minimal locomotion. One of these cases is plotted to show the combination of minimal locomotion vector.

III.3 Assemble a Path

Following Kanayama's method, we adapt two cubic spirals to form a path. There are

three zero curvature points at a two-cubic-spiral path. These points can be extended by straight lines without lose the feature of continuous curvature. Combining with two cubic spirals, we now have five segments to assemble a path to connect a given configuration pair. The five directions are

$$\begin{aligned} & \theta_1, \theta_{c1}, \theta_m, \theta_{c2}, \text{ and } \theta_2, \text{ where} \\ & \theta_{c1} \equiv \Phi\left(\frac{\theta_1 + \theta_m}{2}\right), \text{ and} \\ & \theta_{c2} \equiv \Phi\left(\frac{\theta_m + \theta_2}{2}\right), \end{aligned} \quad (19)$$

Their corresponded vectors are

$$\vec{v}_1, \vec{v}_{c1}, \vec{v}_m, \vec{v}_{c2}, \text{ and } \vec{v}_2 \quad (20)$$

Five vectors are all adjustable; a successful combination of these vectors is to fulfill the below equation,

$$[q_2] = [q_1] + \vec{v}_1 + \vec{v}_{c1} + \vec{v}_m + \vec{v}_{c2} + \vec{v}_2 \quad (21)$$

Since the directions of the five vectors are known, we need only to solve their length.

Define unit vectors of the above five vectors as

$$\vec{n}_1, \vec{n}_{c1}, \vec{n}_m, \vec{n}_{c2}, \text{ and } \vec{n}_2 \quad (22)$$

The positive direction of these five unit vectors are forward motion direction. Then (21) becomes

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

where $d_1, d_{c1}, d_m, d_{c2},$ and d_2 are length to be decided.

By (16), the constraints are

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

$d_1, d_m,$ and d_2 could be positive or negative. We must emphasis again that positive value of each d denotes forward motion and negative does backward one.

III.4 Criterion of Minimal Length

The objective of this section is to find a minimal length solution. Assume θ_m and $i, j, k,$ and l have been selected already. The cost criterion can be defined as

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

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

$d_1, d_{c1}, d_m, d_{c2},$ and d_2 should be solved by (23) under the constraint (24) and to

minimize the above cost criterion. Because d_{c1}^{ij} and d_{c2}^{kl} have been chosen,

$D(\alpha_{c1}^i)$ and $D(\alpha_{c2}^k)$ are constant now. The cost of minimal locomotion is

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

This part of cost cannot be reduced. We define

$$\begin{aligned} d_{c1}^* &= d_{c1} - d_{c1}^{ij} \\ d_{c2}^* &= d_{c2} - d_{c2}^{kl} \end{aligned} \quad (27)$$

Then to minimize (25) is equivalent to minimize

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

And the constraints (24) become

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

III.5 Vector Choose

Substitute (27) to (23), we have

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

Define

$$\bar{v}_{goal} = [q_2] - [q_1] - (d_{c1}^{ij} \bar{n}_{c1} + d_{c2}^{kl} \bar{n}_{c2}) \quad (31)$$

where $d_{c1}^{ij} \bar{n}_{c1} + d_{c2}^{kl} \bar{n}_{c2}$ is the minimal locomotion vector $\bar{v}_c^{ij,kl}$ solved in (18). Then the equation to solve becomes

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

$d_1, d_m,$ and d_2 can be positive or negative, but the signs of d_{c1}^* and d_{c2}^* are pre-decided by (29). It is not convenient to solve such a problem. Therefore we define

$$\vec{n}_\Delta^+ = \vec{n}_\Delta \text{ and } \vec{n}_\Delta^- = -\vec{n}_\Delta \quad (33)$$

$\Delta \in \{1, c1, m, c2, 2\}$. Define a vectors set \mathbb{N} ,

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

j and l is + or -, and its corresponded coefficient set \mathbb{C} ,

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

Then (32) becomes

$$\vec{v}_{goal} = d_1^+ \vec{n}_1^+ + d_1^- \vec{n}_1^- + d_{c1}^j \vec{n}_{c1}^j + d_m^+ \vec{n}_m^+ + d_m^- \vec{n}_m^- + d_{c2}^l \vec{n}_{c2}^l + d_2^+ \vec{n}_2^+ + d_2^- \vec{n}_2^- \quad (36)$$

where

$$\begin{aligned} d_\Delta &= d_\Delta^+ - d_\Delta^-, \Delta \in \{1, m, 2\}, \text{ and} \\ d_{c1}^j &= |d_{c1}^*| \\ d_{c2}^l &= |d_{c2}^*| \end{aligned} \quad (37)$$

Then all unknowns are positive, and we have total eight unknowns to be solved. The criterion (28) becomes

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

The solution is not as hard as it looks like. In fact, to minimize (38) we need only two terms in (35), and let others remain zero. The reason will be addressed in appendix 1. Assume now we have chosen a pair of vector \vec{n}_a and $\vec{n}_b \in \mathbb{N}$, and the corresponded d_a and $d_b \in \mathbb{C}$, (36) becomes

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

By Appendix B, the solution of d_a and d_b is

$$\begin{aligned} d_a &= \|\vec{v}_{goal}\| \frac{-\sin \theta^-}{\sin(\theta^+ - \theta^-)} \\ d_b &= \|\vec{v}_{goal}\| \frac{\sin \theta^+}{\sin(\theta^+ - \theta^-)} \end{aligned} \quad (40)$$

where

$$\begin{aligned} \theta^+ &= \Phi(\text{angle}(\vec{n}_a) - \text{angle}(\vec{v}_{goal})) \\ \theta^- &= \Phi(\text{angle}(\vec{n}_b) - \text{angle}(\vec{v}_{goal})) \end{aligned} \quad (41)$$

The choose of \vec{n}_a and \vec{n}_b will let θ^+ be positive and θ^- negative. Then, by (39),(40),

and (41), coefficients $d_1, d_{c1}, d_m, d_{c2},$ and d_2 is solved and (25) is minimized. Note because d_{c1}^{ij} and d_{c2}^{kl} is usually non-zero, the amounts of non-zero coefficients are between two and four, i.e., there is at least one zero coefficient.

III.6 Procedure to Find a Best Path

The procedure to find a best path is to search all possible combination of cubic spiral pairs, and find the one that minimizes the cost criterion (25). From previous section, we have found the shortest path of a specific θ_m and one of its $\vec{v}_c^{ij,kl}$. To find the best path, we should compute the cost of each case that θ_m from $-\pi$ to π and its sixteen $\vec{v}_c^{ij,kl}$. Though there may be thousands of cases to check, the task is in fact not so hard. This is because we do not have to compute each cubic spiral data to measure the path length, but only have to generate five d to compute the cost. The Procedure is

$$\begin{aligned}
& \text{for } \theta_m = -\pi \text{ to } \pi \text{ step } \Delta\theta \\
& \quad \text{for } i = +, - \quad (\alpha_{c1}^i = \alpha_{c1}^-, \alpha_{c1}^+) \\
& \quad \quad \text{for } j = +, - \quad (\text{Direction } C_1 = \text{Forward}, \text{Backward}) \\
& \quad \quad \quad \text{for } k = +, - \quad (\alpha_{c2}^k = \alpha_{c2}^-, \alpha_{c2}^+) \\
& \quad \quad \quad \quad \text{for } l = +, - \quad (\text{Direction } C_2 = \text{Forward}, \text{Backward}) \\
& \quad \quad \quad \quad \quad \text{Compute_Cost} \tag{42} \\
& \quad \quad \quad \quad \quad \text{next} \\
& \quad \quad \quad \quad \text{next} \\
& \quad \quad \quad \text{next} \\
& \quad \quad \text{next} \\
& \quad \text{next}
\end{aligned}$$

$\Delta\theta$ can be chosen according to requirements, as we will further discuss later. Some paths planned by the algorithm are shown in Fig. 5 and 6. Fig. 7 is an example to represent the procedure to optimize.

IV Further Discussion

Some important feather of this path planning method is discussed in this section.

IV.1 Forward and Backward Motion

The path planning algorithm suggested in this paper is for the vehicle which can drive both forward and backward. However, it also can be applied to the uni-direction path

planning problem. We should simply restrict *Direction* C_1 and *Direction* C_2 be forward, and use

$$\begin{aligned}\mathbb{N}_f &\equiv \{\vec{n}_1^+, \vec{n}_m^+, \vec{n}_2^+, \vec{n}_{c1}^+, \vec{n}_{c2}^+\} \\ \mathbb{C}_f &\equiv \{d_1^+, d_m^+, d_2^+, d_{c1}^+, d_{c2}^+\}\end{aligned}\quad (43)$$

to replace \mathbb{N} and \mathbb{C} . Then the solution can fit our use. But one should note that when solve (39), the requirement for positive solution (d_a and d_b) is

$$\theta^+ < \frac{\pi}{2} \text{ and } \theta^- > -\frac{\pi}{2} \quad (44)$$

In the forward motion case, for some θ_m and $\vec{v}_c^{ij,kl}$, such a pair of (\vec{v}_a, \vec{v}_b) may not exist, then the algorithm is failure to generate a path. Its cost is set to be an extreme large value to avoid being chosen as solution. Fig. 8 and 9 show some uni-direction path planned by this algorithm.

IV.2 Wall-Collision Avoidance

For practical use a vehicle path usually has more constraints, because the space is not obstacle-free or is restricted by some walls. A complete path planning method should also consider these restrictions. Our revised path planning method can generate many acceptable paths speedily. Though in previous section we only choose the shortest path to be the best one, this does not means other paths are not usable. Here we introduce another term in the cost function to simply avoid wall-collision.

There are six points along a planned path can be computed when \mathbb{C} is solved; they are

$$\begin{aligned}p_0 &= [q_1] \\ p_1 &= p_0 + \vec{v}_1 \\ p_2 &= p_1 + \vec{v}_{c1} \\ p_3 &= p_2 + \vec{v}_m \\ p_4 &= p_3 + \vec{v}_{c2} \\ p_5 &= p_4 + \vec{v}_2 = [q_2]\end{aligned}\quad (45)$$

where $p_i \equiv (p_{ix}, p_{iy})$ is a coordinate in \mathbb{R}^2 . Some of these points may be identical, because some d are zero. A wall-type obstacle is modeled as a straight line equation in \mathbb{R}^2 space,

$$mx + c_0 - y = 0 \quad (46)$$

Points p_0 and p_5 are automatically collision-free, so they must lie on the same side of above straight line equation. Therefore, define function $O(p)$ as,

$$O(p) \equiv mp_x + c_0 - p_y \quad (47)$$

Then if

$$O(p_0) \cdot O(p_i) > 0, \text{ for } i = 1, 2, 3, 4 \quad (48)$$

Then we can conclude that all five points are collision-free. For a (46) type obstacle and a straight line segment of a path, its two end points collision free means this segment is collision free. However for a cubic spiral segment, this may not be true. Therefore we should check more points to make sure the safety. There is no easy way to compute the coordinate of selected points on a cubic spiral without using equation (5). But the integration costs too much computation time. Therefore we implement a simplified method to generate check points, as described below.

For a cubic spiral segment, we design to add two more check points. We generate three straight line segments to fit a cubic spiral one, and use the two more vertex as check points. The total lengths of these three straight line segments are equal to the length of cubic spiral. As shown in Figure 10, assume a cubic spiral has size d_{c1} and direction \vec{v} . The second portion is parallel to the \vec{v} , and its length is h_{c1} , we have

$$\begin{aligned} \frac{d_{c1}}{D(\alpha)} &= h_{c1} + |d_{c1} - h_{c1}| \cdot S(\alpha), \text{ where} \\ S(\alpha) &\equiv \frac{2 \sin(\frac{\alpha}{2})}{\sin(\alpha)} \end{aligned} \quad (49)$$

Rearrange above equation, we obtain

$$\frac{h_{c1}}{d_{c1}} = \frac{1}{1 - \text{sgn}(d_{c1} - h_{c1}) \cdot S(\alpha)} - \text{sgn}(d_{c1} - h_{c1}) \cdot S(\alpha) \quad (50)$$

Lengths of the other two line segments are,

$$l_{c1} = \frac{|d_{c1} - h_{c1}|}{2 \cos(\alpha/2)} \quad (51)$$

From fig 10, the two more check points can be solved by vector addition.

$$\begin{aligned} p_1^* &= p_1 + l_{c1} \cdot \vec{n}_1 \\ p_2^* &= p_2 - l_{c1} \cdot \vec{n}_m \end{aligned} \quad (52)$$

These two points are close to the original cubic spiral, and check them can be a good approximation of collision detection of this segment. Furthermore, one may note that this two more check points are collinear with the two straight line segments that connect with this cubic spiral. For a line type segment we need only to check its two end points to know whether the collision happens or not. Therefore, finally the check points remain five or less than five. To use the points of (52) or to the original ones is dependent on the motion direction change of the path.

We can simply set $\text{cost}(\Pi)$ to be an extreme large value to avoid choosing path that collision will happen, i.e. some check points that let (48) be failure to hold. This method is effective and can eliminate most unqualified trajectories. Some examples are shown in Fig 11.

IV.3 An Effective Search

In (42) if we set $\Delta\theta = 0.0873$, which is approximately equal to 5 degree, then we should search 1152 cases to find a best one. But in the forward-and-backward and free space case one can found a cubic spiral which angle outside the range $[-\pi, \pi]$ seldom appears in the result path. We shall explain this and try to find a more efficient procedure to search an acceptable solution.

The abstract of the explanation is that first we will assume a result path includes a cubic spiral with angle outside the range $[-\pi, \pi]$, and then prove that in most cases there exists at least one shorter path and the shorter one will replace the original one due to the linear programming optimization. Following are the details.

Assume in a path optimized by the method described in previous sections there is one cubic spiral with an angle outside the range $[-\pi, \pi]$. By our definition, its angle α has a superscript “-“, i.e. α^- . And the following equation must hold

$$d \geq d_{\min}(\alpha^-) \quad (53)$$

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

$$\ell_{\alpha^-} = \frac{d}{D(\alpha^-)} \quad (54)$$

Because the corresponded angle α^+ inside the range $[-\pi, \pi]$ also steers the vehicle to the same direction, we can also build another path by α^+ to substitute this one without contradict the continuous curvature constraint. Following are two cases:

Case 1, $d \geq d_{\min}(\alpha^+)$

In this case an alternate path can be constructed by a single cubic spiral with angle α^+ and size d , the path length is

$$\ell_{\alpha^+} = \frac{d}{D(\alpha^+)} \quad (55)$$

By Fig. 1, we can directly figure that for any two angle β_1 and β_2 ,

$$\begin{aligned} D(\beta_1) &> D(\beta_2) > 0, \\ \text{if } |\beta_1| < \pi \text{ and } |\beta_2| > \pi \end{aligned} \quad (56)$$

which means

$$\frac{d}{D(\alpha^+)} < \frac{d}{D(\alpha^-)}, i.e. \quad (57)$$

$$\ell_{\alpha^+} < \ell_{\alpha^-}$$

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

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

$$\ell_{\alpha^+} = \frac{d_{\min}(\alpha^+)}{D(\alpha^+)} + (d_{\min}(\alpha^+) - d) \cdot \frac{2 \sin(\frac{\alpha^+}{2})}{\sin(\alpha^+)} \quad (58)$$

Then

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

$$\frac{d_{\min}(\alpha^+)}{D(\alpha^+)} + (d_{\min}(\alpha^+) - d) \cdot \frac{2 \sin(\frac{\alpha^+}{2})}{\sin(\alpha^+)} - \frac{d}{D(\alpha^-)} \geq 0 \text{ holds} \quad (59)$$

The above condition can be rearranged as

$$\frac{\frac{1}{D(\alpha^+)} + S(\alpha^+)}{\frac{1}{D(\alpha^-)} + S(\alpha^+)} \geq \frac{d}{d_{\min}(\alpha^+)} \quad (60)$$

Multiply both side of (56) by $\frac{d_{\min}(\alpha^+)}{d_{\min}(\alpha^-)}$, we get

$$\frac{\frac{1}{D(\alpha^+)} + S(\alpha^+)}{\frac{1}{D(\alpha^-)} + S(\alpha^+)} \cdot \frac{d_{\min}(\alpha^+)}{d_{\min}(\alpha^-)} \geq \frac{d}{d_{\min}(\alpha^-)} \quad (61)$$

Fig. 13 shows the value of LHS of (61). By the assumption, RHS of (61) must be larger or equal than 1, but in the figure, it is easy to point that when $|\alpha^+| < 139^\circ$, all

values of LHS are less than 1. Furthermore only when $\frac{d}{d_{\min}(\alpha^-)} < 1.0734$, (61) is

possible to hold. The gray region on the figure is where $\ell_{\alpha^+} \geq \ell_{\alpha^-}$ holds. Apparently it

is small; so that the condition that cubic spiral with angle α^- cannot be replaced by corresponded α^+ one seldom happens.

Both on the above two case we can almost find a shorter path to replace the original

cubic spiral with angle α^- . By the shortest path criterion, this original cubic spiral can not appear in the final result if the shorter is found.

This concludes that in a two-direction and no-obstacle path planning problem, we need not to check the α_{c1}^- and α_{c2}^- cases, and can save half of computation time with little optimization sacrifice.

V. Conclusion

In this paper we discuss an improved path planning method using both cubic spiral and straight line segments. A planned path is constituted by at most two cubic spirals and two straight lines, and can be assured the shortest one of this combination. The improved method is suitable for various vehicles or mobile robots' path planning because it considers the bounded curvature constraint and can generate paths with or without cusp, i.e., direction change. Moreover, the continuous curvature of generated path makes controller design easier. The complete method searches all possible paths via the change of intermediate configuration, but we also show that a faster search is achievable with few optimization loss. As for practical use, a mobile robot usually executes its task in a constraint workspace, so a simple wall collision avoidance scheme is discussed. It checks a few points of the path and excludes dangerous ones. This method is being applied to a robot soccer game project that we design a ball passing strategy and use multiple mobile robots to realize it by change their formation dynamically. It works well.

Appendix

A Minimal Length

If we let

$$\begin{aligned}
 x &= [d_1^+, d_1^-, d_m^+, d_m^-, d_2^+, d_2^-, d_{c1}^j, d_{c2}^l]^T \\
 A &= \begin{bmatrix} \cos\theta_1 & -\cos\theta_1 & \cos\theta_m & -\cos\theta_m & \cos\theta_2 & -\cos\theta_2 & \text{sgn}(d_{c1}^j) \cdot \cos\theta_{c1} & \text{sgn}(d_{c2}^l) \cdot \cos\theta_{c2} \\ \sin\theta_1 & -\sin\theta_1 & \sin\theta_m & -\sin\theta_m & \sin\theta_2 & -\sin\theta_2 & \text{sgn}(d_{c1}^j) \cdot \sin\theta_{c1} & \text{sgn}(d_{c2}^l) \cdot \sin\theta_{c2} \end{bmatrix} \\
 b &= \vec{v}_{goal} \\
 c^T &= [-1, -1, -1, -1, -1, -1, -\frac{1}{D(\alpha_{c1}^j)}, -\frac{1}{D(\alpha_{c2}^k)}]
 \end{aligned} \tag{A1}$$

Then the original problem becomes a canonical form of linear programming problem,

$$\begin{aligned}
 &\text{maximize} && -\text{cost}^*(\Pi) = c^T x \\
 &\text{subject to} && Ax = b \\
 &&& x \geq 0
 \end{aligned} \tag{A2}$$

which can be solved by simplex method suggested by George B. Dantzig in 1947[11].

Because one of the extreme point is the optimal solution, we can directly know that the optimal solution of x has only two non-zero elements x_a and x_b . A reduced vector is defined as

$$x_r = [x_a, x_b] \quad (A3)$$

and the corresponded columns of matrix A is also collected,

$$A_r = [A_a | A_b] \quad (A4)$$

Then the below equation also holds

$$A_r x_r = \vec{v}_{goal} \quad (A5)$$

This equation is equivalent to (39) if let $d_a = x_a$ and $d_b = x_b$.

The number of iterations is of order 2 to 4 times the number of rows of matrix A , and the total running time has been found roughly to the equation

$$Time = K \times (No. \text{ of Rows of } A)^3 \quad (A6)$$

However, due to the specific geometric relations, we do have a faster method to find which two constraints, i.e. which two rows of A , forms the extreme point. The method is described here.

A.2 A Faster Method

Let

$$\Theta \equiv \{Angle(angle(\vec{n}) - angle(\vec{v}_{goal})) : \vec{n} \in \mathbb{N}\} \quad (A7)$$

Then let

$$\begin{aligned} \theta^{mp} &= \text{the minimal positive element of } \Theta \\ \theta^{Mn} &= \text{the maximal negative element of } \Theta \\ \theta^{vmp} &= \text{the vice minimal positive element of } \Theta \\ \theta^{vMn} &= \text{the vice maximal negative element of } \Theta \end{aligned} \quad (A8)$$

If the corresponded \vec{n} s of both θ^{mp} and θ^{Mn} are not \vec{n}_{c1}^j or \vec{n}_{c2}^l , then

$$\theta^+ = \theta^{mp} \text{ and } \theta^- = \theta^{Mn} \quad (A9)$$

Else if either the corresponded θ^{mp} or θ^{Mn} is \vec{n}_{c1}^j or \vec{n}_{c2}^l , then we should check θ^{vmp} and θ^{vMn} to compare which solution is maximal. Two more combinations of solutions could exist,

$$\begin{aligned} \theta^+ &= \theta^{vmp} \text{ and } \theta^- = \theta^{Mn}, \text{ or} \\ \theta^+ &= \theta^{mp} \text{ and } \theta^- = \theta^{vMn} \end{aligned} \quad (A10)$$

Finally if corresponded \vec{n} s of both θ^{mp} and θ^{Mn} are \vec{n}_{c1}^j or \vec{n}_{c2}^l , one more combination of solution should be check,

$$\theta^+ = \theta^{vmp} \text{ and } \theta^- = \theta^{vMn} \quad (\text{A11})$$

Therefore, we should check at most four combinations of solutions to find the extreme point.

B. Notes of $S(\alpha)$

As shown in Fig.14, use of two unit vectors \vec{n}_1 and \vec{n}_2 to represent a known vector \vec{v}_0 usually happens in our discussion. The angle between \vec{n}_1 and \vec{v}_0 is γ_1 ; \vec{v}_0 and \vec{n}_2 is γ_2 . Two lengths d_1 and d_2 can be solved by,

$$\begin{bmatrix} v_{0x} \\ v_{0y} \end{bmatrix} = \begin{bmatrix} \cos \gamma_1 & \cos \gamma_2 \\ \sin \gamma_1 & \sin \gamma_2 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (\text{B1})$$

The solution of above equation can be obtained by

$$\begin{aligned} d_1 &= \|\vec{v}_0\| \frac{-\sin(\gamma_2)}{\sin(\gamma_1 - \gamma_2)} \\ d_2 &= \|\vec{v}_0\| \frac{\sin(\gamma_1)}{\sin(\gamma_1 - \gamma_2)} \end{aligned} \quad (\text{B2})$$

If $\gamma_2 = -\gamma_1$, we have

$$d_1 = d_2 = \|\vec{v}_0\| \frac{\sin(\gamma_1)}{\sin(2\gamma_1)} \quad (\text{B3})$$

and

$$d_1 + d_2 = \|\vec{v}_0\| \frac{2\sin(\gamma_1)}{\sin(2\gamma_1)} \quad (\text{B4})$$

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

References

- [1] Y. J. Kanayama and B. I. Hartman, "Smooth Local Path Planning for Autonomous Vehicles," *The International Journal of Robotics Research*, vol .16, no. 3, June 1997, pp. 263-283.
- [2] P. Souères and J. Laumond, "Shortest Path Synthesis for a Car-Like Robot," *IEEE Transactions on Automatic Control*, vol. 41, no. 5, May 1996, pp. 672-688.
- [3] A. Scheuer and Ch. Laugier, "Planning Sub-Optimal and Continuous-Curvature Paths for Car-Like Robots," Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B. C., Canada, October 1998.
- [4] A. Scheuer and Th. Fraichard, "Collision-Free and Continuous-Curvature Path

Planning for Car-Like Robots,” *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 867-873

[5] A. M. Shkel and V. J. Lumelsky, “On Calculation of Optimal Paths with Constrained Curvature: The Case of Long Paths,” *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 3578-3583.

[6] A. M. Shkel and V. J. Lumelsky, “Curvature-Constrained Motion Within a Limited Workspace,” *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 1394-1399.

[7] J. Boissonnat, A. Cérézo, and J. Leblond, “Shortest Paths of Bounded Curvature in the Plane,” *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 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,” *American Journal of Mathematics*, vol. 79, pp. 497-516, 1957.

[9] J. A. Reeds and R. A. Shepp, “Optimal Paths for a Car that Goes Both Forward and Backwards,” *Pacific Journal of Mathematics*, vol. 145, no. 2, 1990.

[10] Y. J. Kanayama and N. Miyake, “Trajectory Generation for Mobile Robots,” *Robotic Research*, vol. 3, Cambridge, MA: MIT Press, 1986, pp. 333-340.

[11] B. Kolman and R. E. Beck, *Elementary Linear Programming with Applications*, Academic Press, New York, 1980

Fig. 1 Distance function of cubic spirals. The dashed line represents $|\alpha| = 180^\circ$; left

side of this line is α^+ region and right side is α^- region [1].

Fig. 2 Two paths planned by Kanayama’s cubic spiral method.

Fig. 3 Four trajectories from an initial configuration q_1 to an intermediate direction (q_m) with maximal (or minimal) curvature value at their middle points.

Fig. 4 Sixteen different end positions of q_1 through minimal locomotions. The cubic spirals of one of them is plotted to show the combination of minimal locomotion vector. ($i: +, j: +, k: +, \text{and}, l: -$)

Fig. 5 Selected path planning results. Each path has the same (q_2) but different [q_2].

Fig. 6 Selected path planning results. Each path has the same [q_2] but different (q_2).

Fig. 7 This figure shows a example of finding an shortest path if θ_m and $i, j, k, \text{and } l$

have been decided. First we use the minimal locomotion vector $\vec{v}_{c1}^{ij} + \vec{v}_{c2}^{kl}$ to find

the \vec{v}_{goal} . Then two vectors to be extended are selected from eight candidates. In this case, \vec{v}'_{c2} and \vec{v}_1^+ are chosen, so the result path plotted in solid line includes one straight line and two cubic spirals.

Fig. 8 Selected path planning results of uni-direction cases. Each path has the same (q_2) but different $[q_2]$.

Fig. 9 Selected path planning results of uni-direction cases. Each path has the same $[q_2]$ but different (q_2) .

Fig. 10 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 check equation (48), but p_2^* will not replace p_2 . This is because p_2^* lie between p_2 and p_3 , and we need only check two endpoints of a straight line.

Fig. 11 Selected cases to show the wall-collision avoidance path planning results.

Fig. 12 This figure shows the method to generate an alternate path by α^+ cubic spiral to replace α^- one if $d < d_{min}(\alpha^+)$. The α^- path is plotted in dotted line and the alternate path is dashing line, which includes two straight line segments and an α^+ cubic spiral.

Fig. 13 The value of LHS of (61).

Fig. 14 Use two vectors \vec{n}_1 and \vec{n}_2 to represent a known vector \vec{v}_0 . γ_1 and γ_2 are angles between (\vec{n}_1, \vec{v}_0) and (\vec{v}_0, \vec{n}_2) .

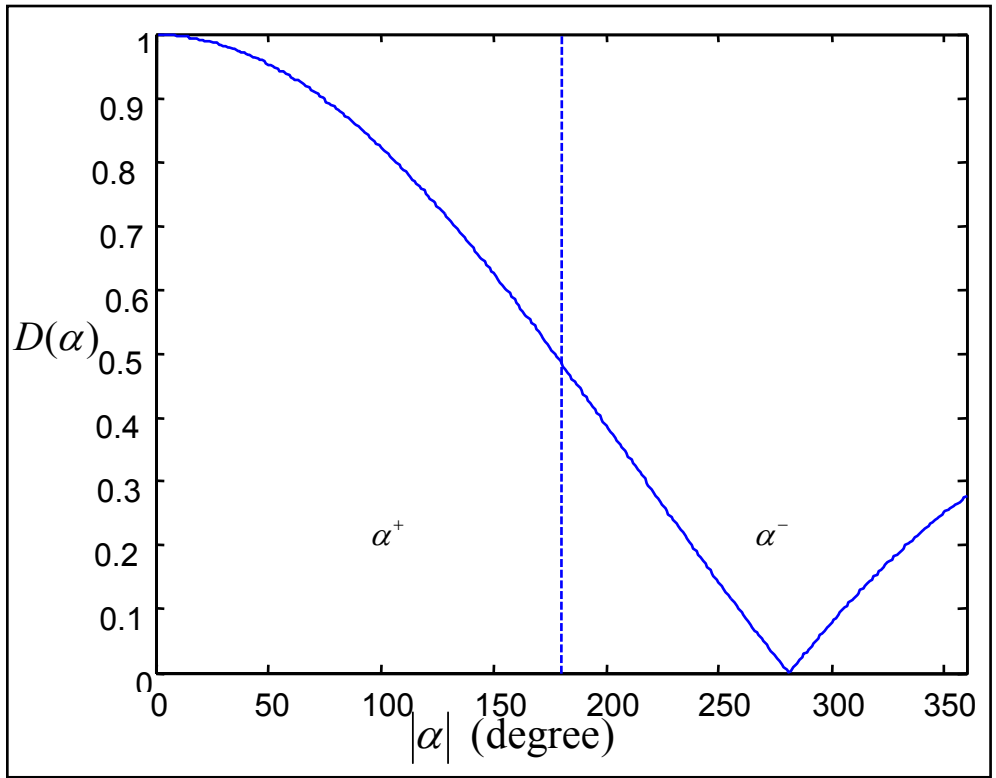


Fig. 1 Distance function of cubic spirals. The dashed line represents $|\alpha|=180^\circ$; left side of this line is α^+ region and right side is α^- region [1].

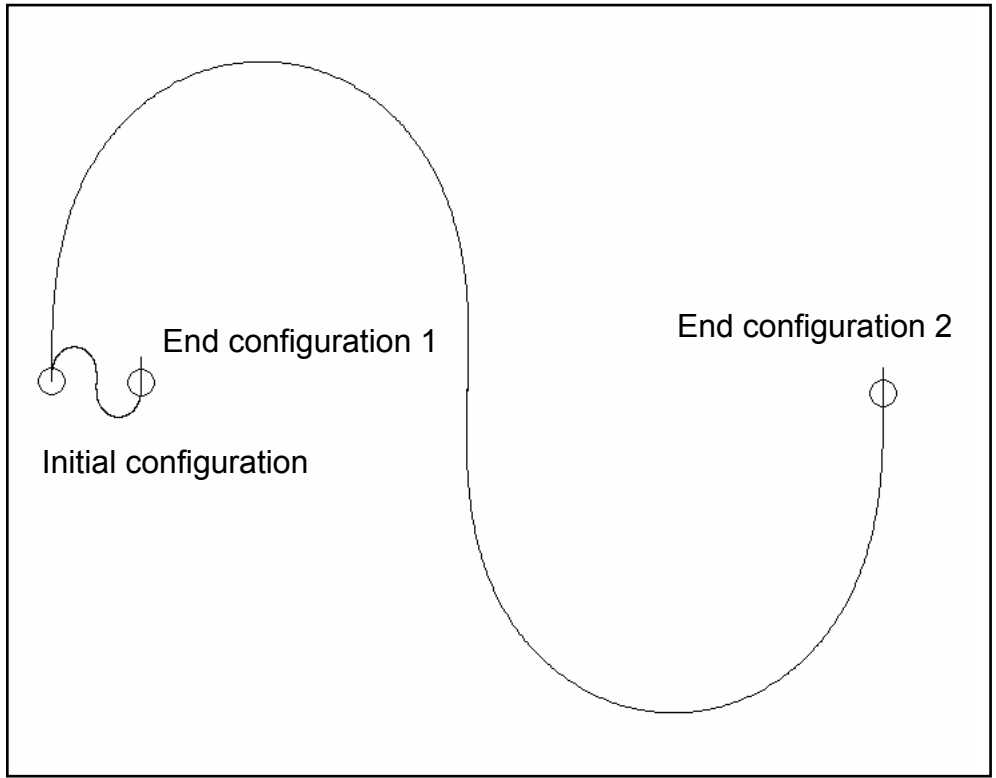


Fig. 2 Two paths planned by Kanayama's cubic spiral method.

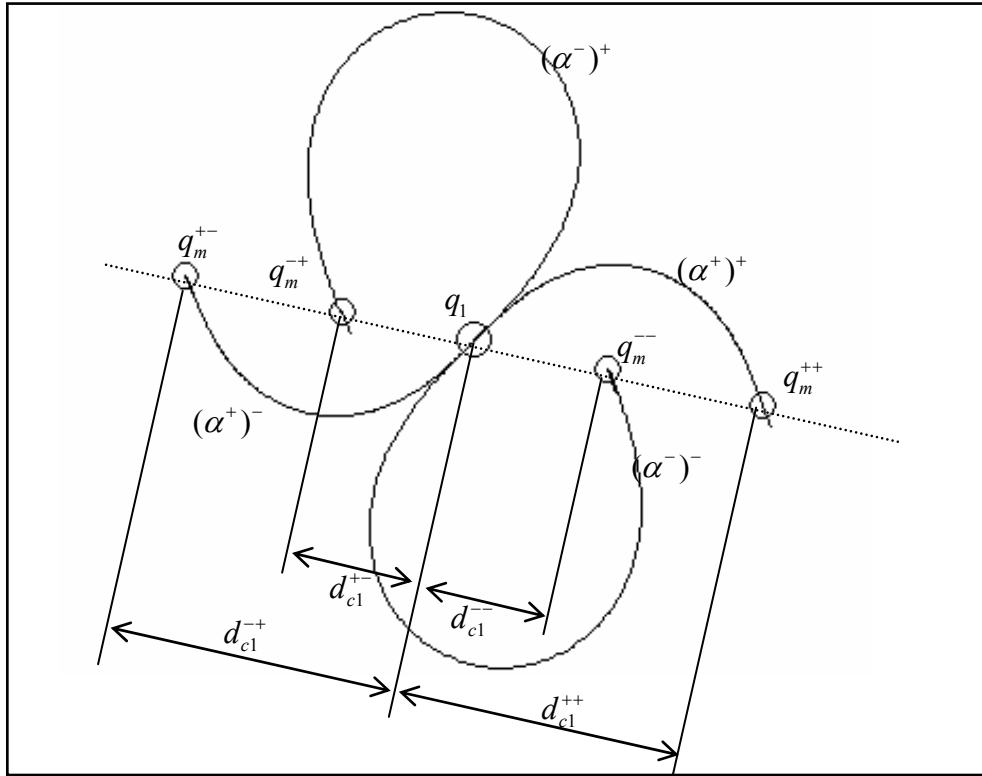


Fig. 3 Four trajectories from an initial configuration q_1 to an intermediate direction (q_m) with maximal (or minimal) curvature value at their middle points.

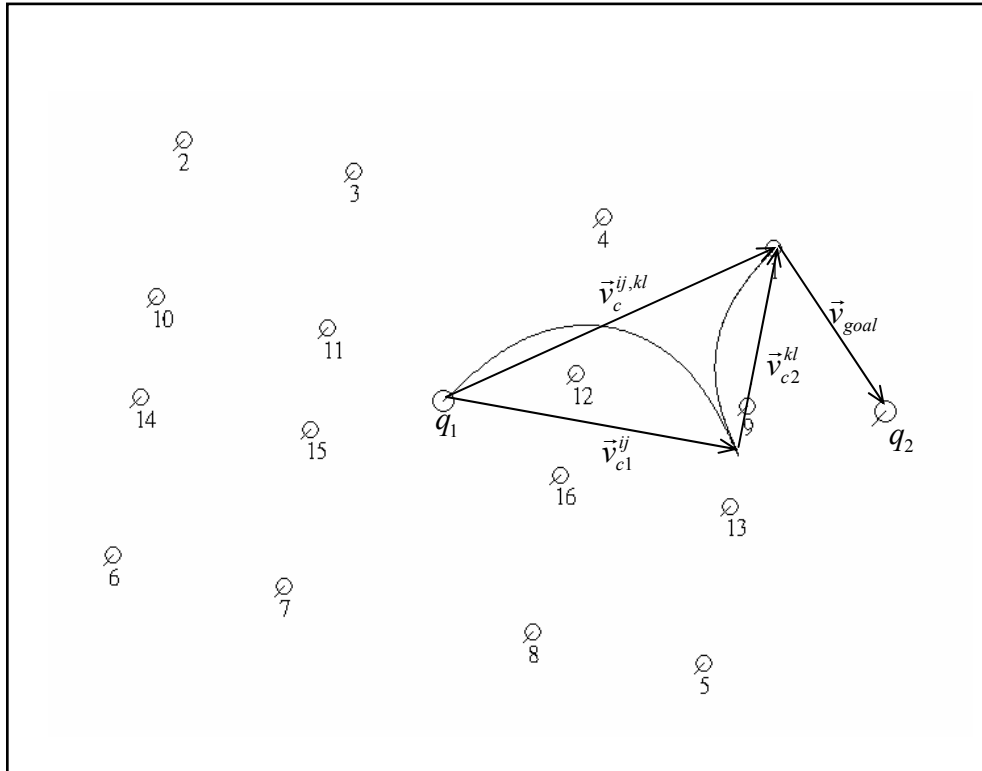


Fig. 4 Sixteen different end positions of q_1 through minimal locomotions. The cubic spirals of one of them is plotted to show the combination of minimal locomotion

vector. ($i:+,j:+,k:+,and,l:-$)

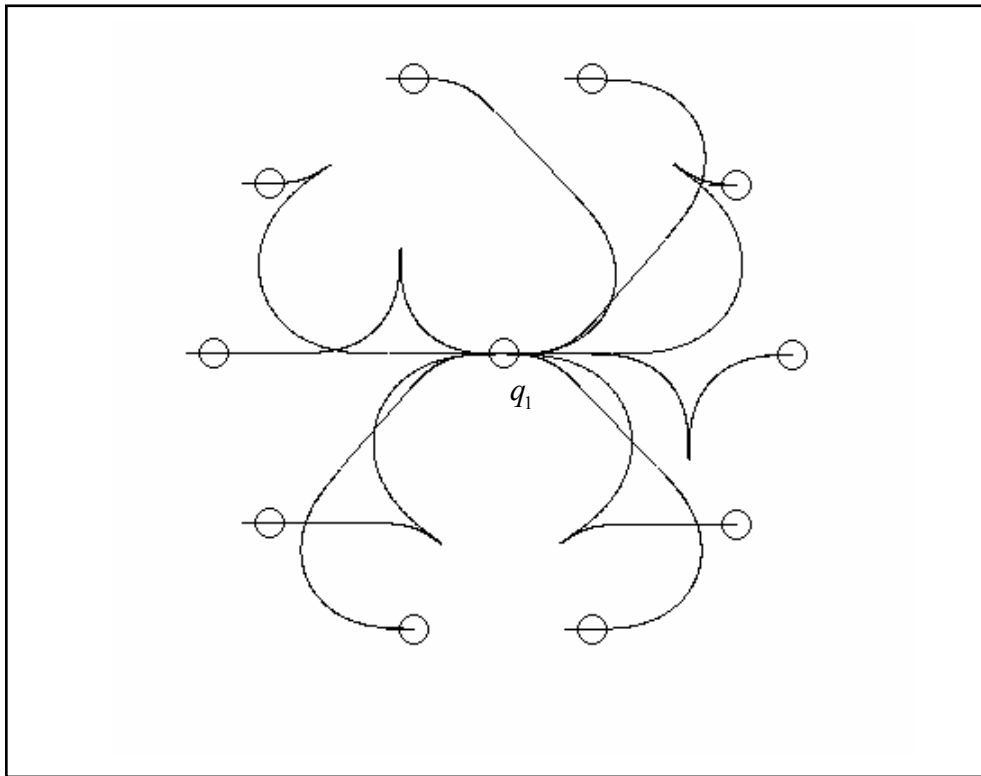


Fig. 5 Selected path planning results. Each path has the same (q_2) but different $[q_2]$.

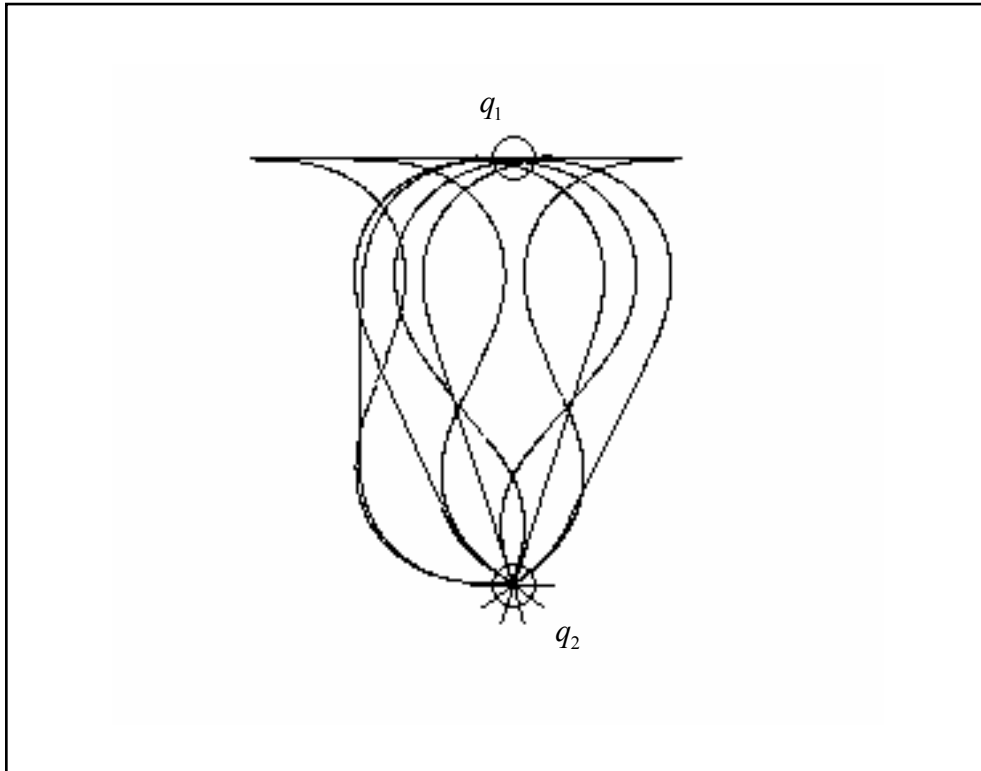


Fig. 6 Selected path planning results. Each path has the same $[q_2]$ but different (q_2) .

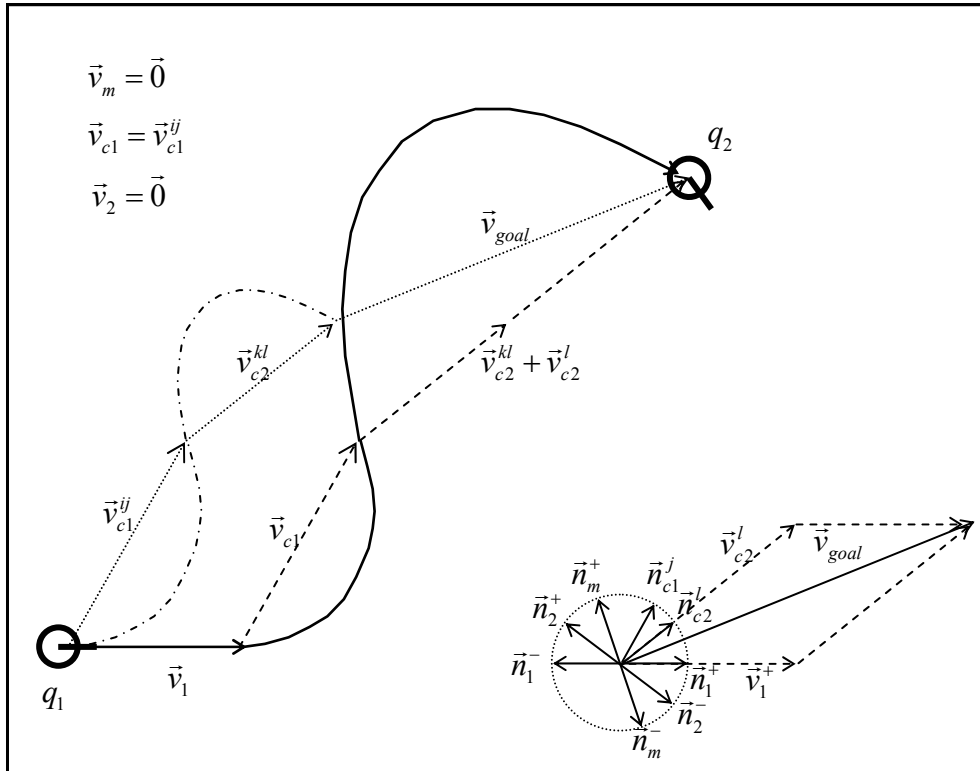


Fig. 7 This figure shows a example of finding an shortest path if θ_m and $i, j, k,$ and l have been decided. First we use the minimal locomotion vector $\vec{v}_{c1}^{ij} + \vec{v}_{c2}^{kl}$ to find the \vec{v}_{goal} . Then two vectors to be extended are selected from eight candidates. In this case, \vec{v}_{c2}^l and \vec{v}_1^+ are chosen, so the result path plotted in solid line includes one straight line and two cubic spirals.

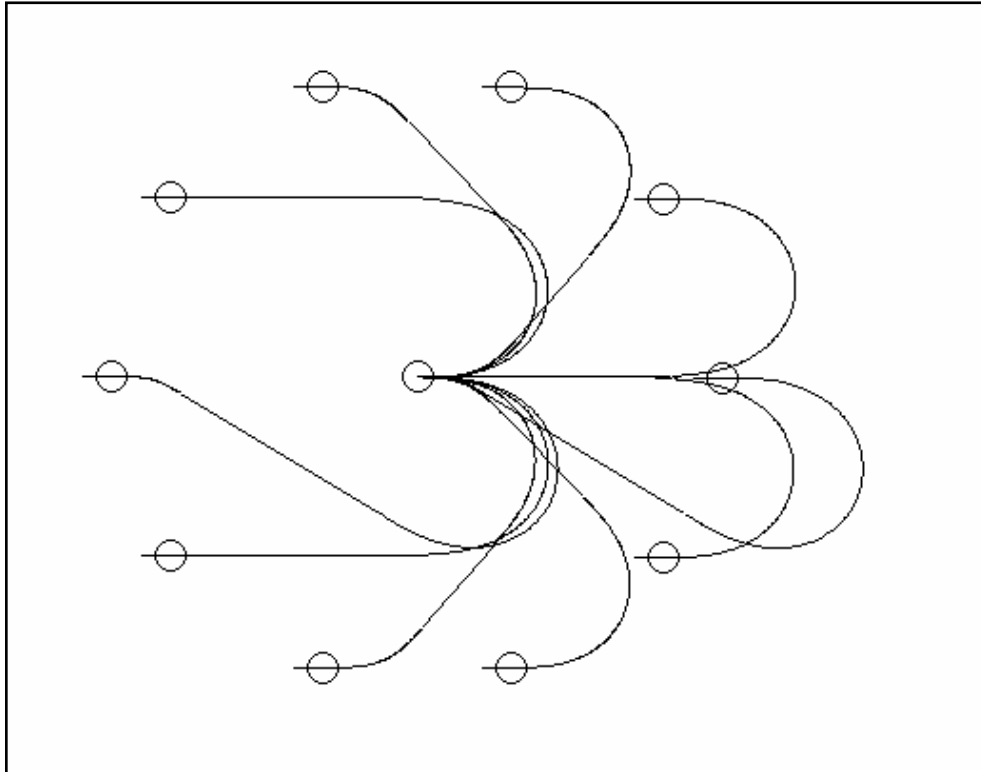


Fig. 8 Selected path planning results of uni-direction cases. Each path has the same (q_2) but different $[q_2]$.

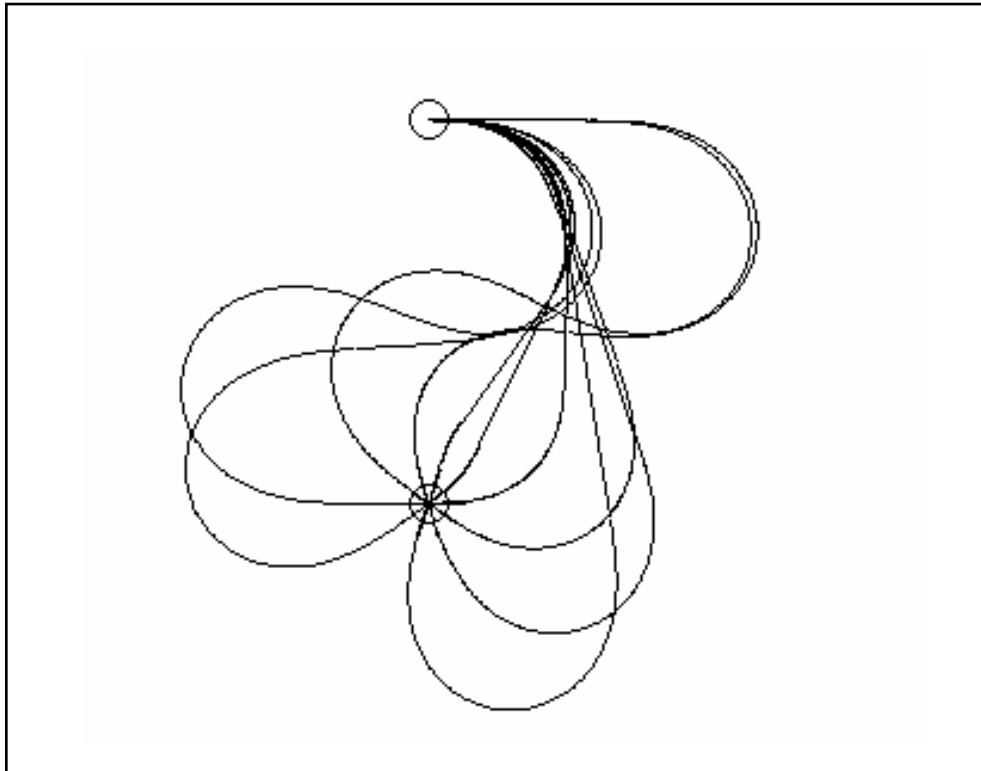


Fig. 9 Selected path planning results of uni-direction cases. Each path has the same $[q_2]$ but different (q_2) .

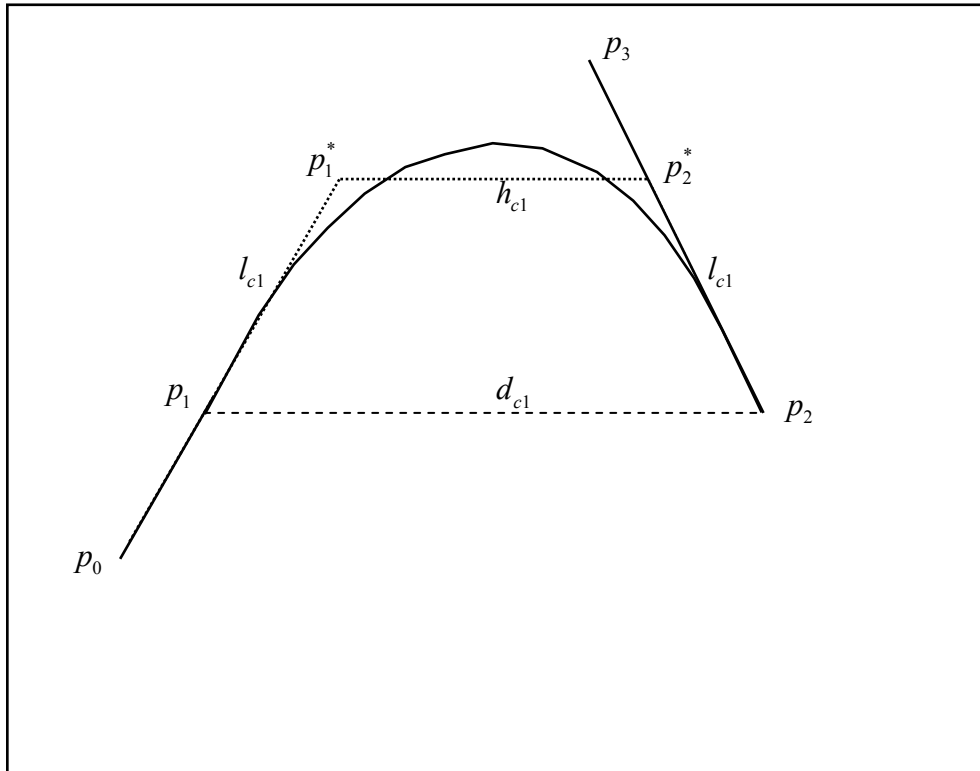


Fig. 10 This figure shows how to generate two more check points p_1^* and p_2^* from p_1 and p_2 . Note that p_1^* will replace p_1 in check equation (48), but p_2^* will not replace p_2 . This is because p_2^* lie between p_2 and p_3 , and we need only check two endpoints of a straight line.

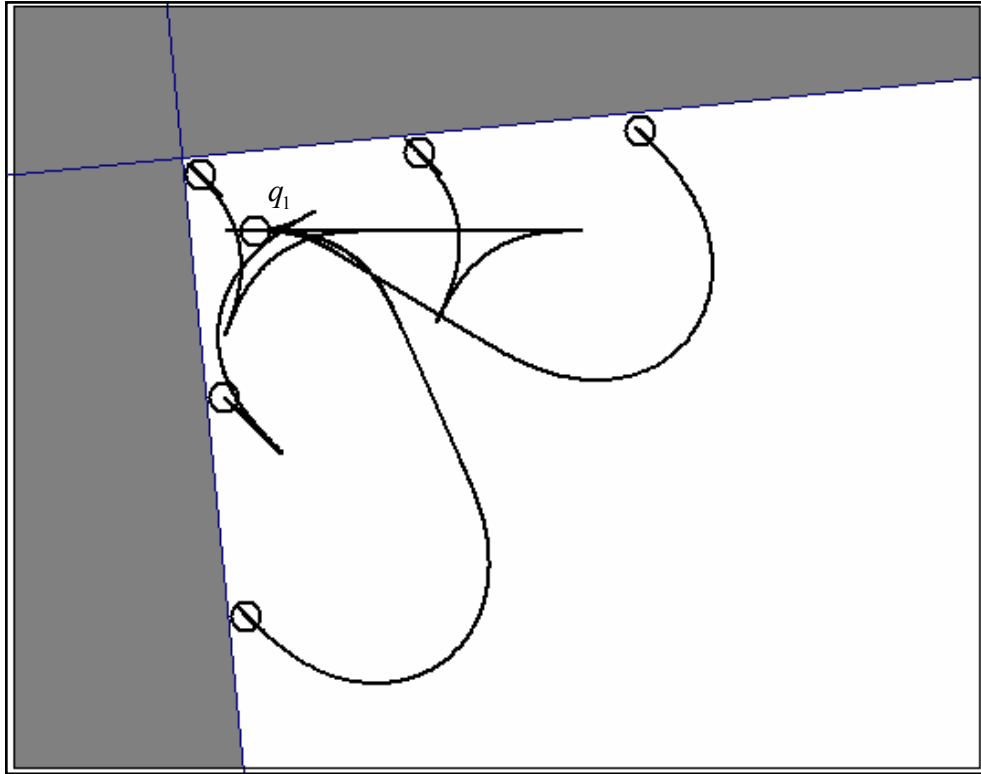


Fig. 11 Selected cases to show the wall-collision avoidance path planning results.

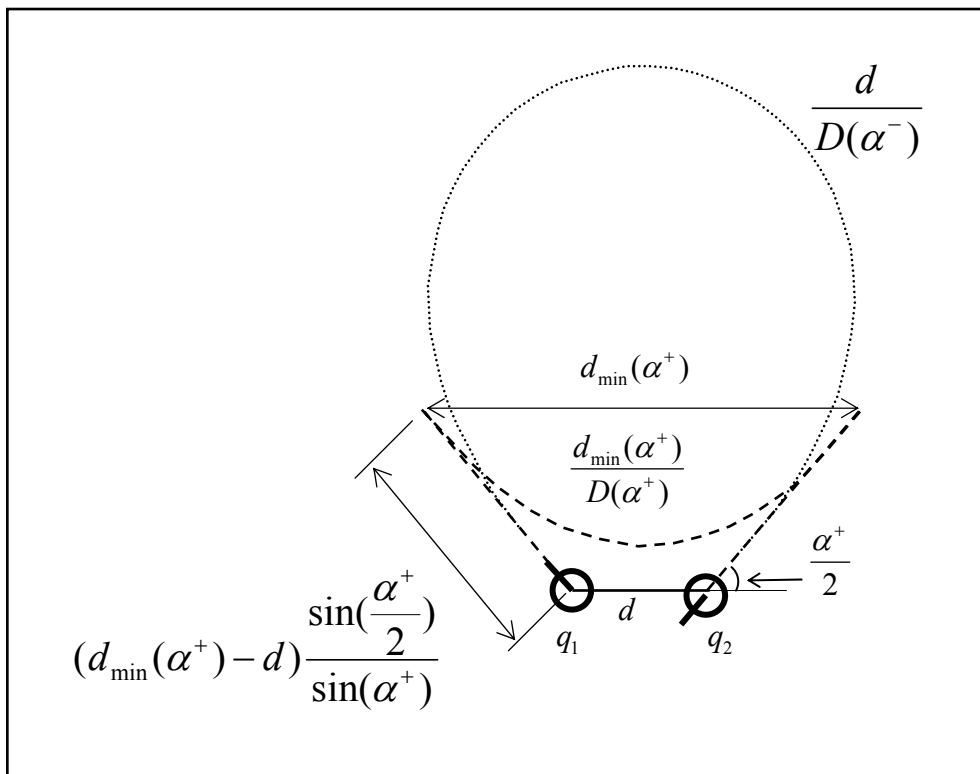


Fig. 12 This figure shows the method to generate an alternate path by α^+ cubic spiral to replace α^- one if $d < d_{\min}(\alpha^+)$. The α^- path is plotted in dotted line and the

alternate path is dashed line, which includes two straight line segments and an α^+ cubic spiral.

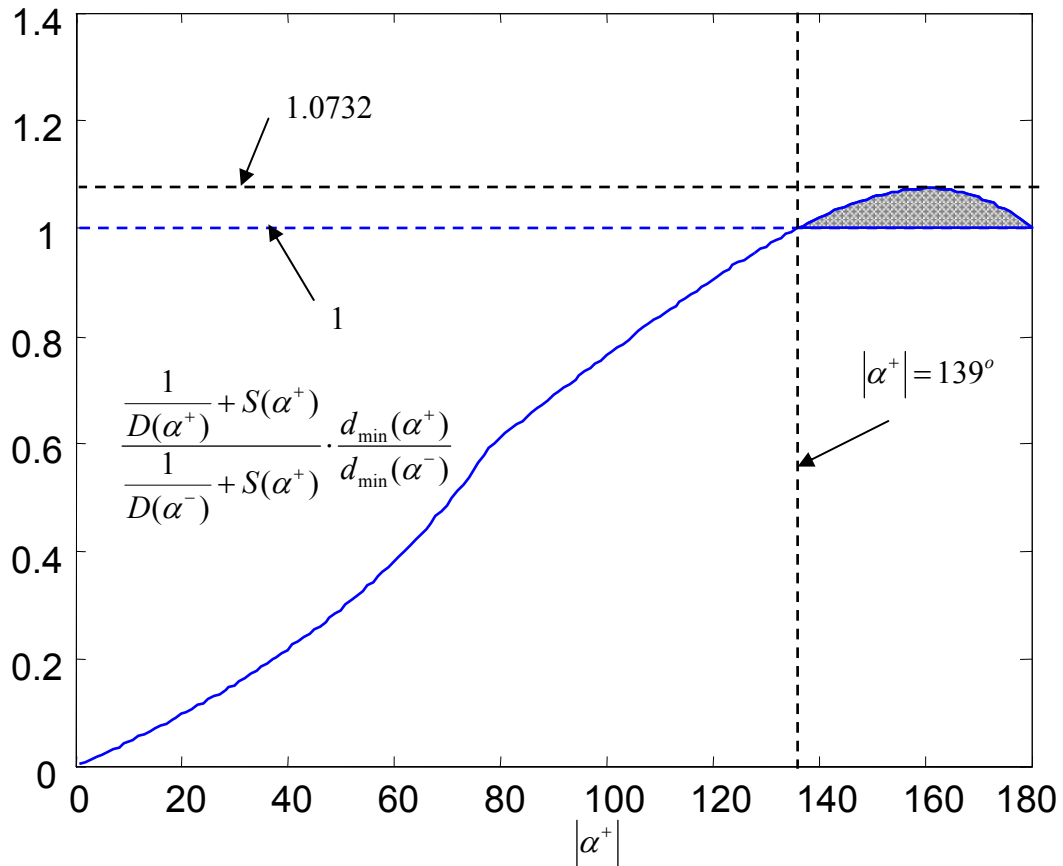


Fig. 13 The value of LHS of (61).

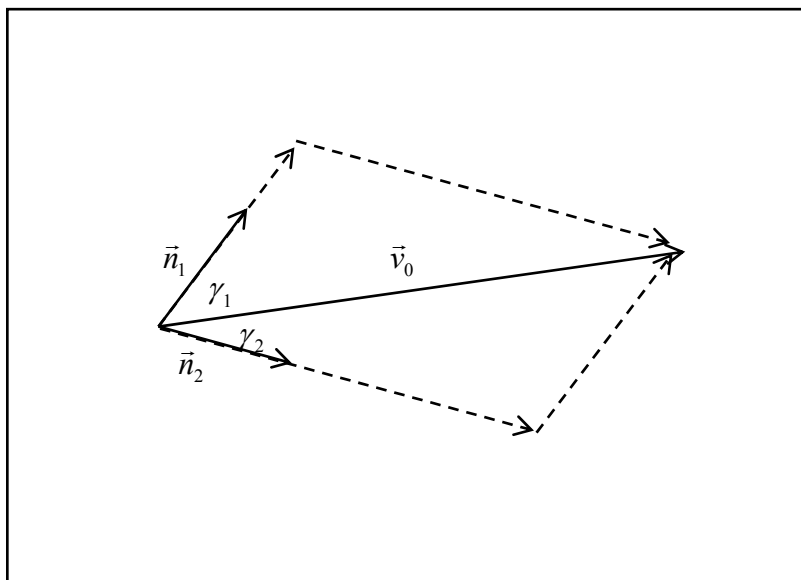


Fig. 14 Use two vectors \vec{n}_1 and \vec{n}_2 to represent a known vector \vec{v}_0 . γ_1 and γ_2 are angles between (\vec{n}_1, \vec{v}_0) and (\vec{v}_0, \vec{n}_2) .