Int. Conf. on Advanced Robotics
October 25-27, 1999. Tokyo (JP), pp. 585–590

From Reeds and Shepp's to Continuous-Curvature Paths

Th. Fraichard, A. Scheuer and R. Desvigne
Inria[a] Rhne-Alpes & Gravir[b]
Zirst. 655 avenue de l'Europe. 38330 Montbonnot Saint Martin. France

December 17, 1999

Abstract — Most path planners for car-like robots compute "Reeds and Shepp paths" made up of line segments connected with circular arcs. Such paths have a discontinuous curvature that makes them difficult to track (curvature is related to the orientation of the front wheels). The purpose of this paper is to present one of the first path planner for car-like robots that computes paths with continuous-curvature and upper-bounded curvature derivative (curvature derivative is related to the steering velocity). The approach presented herein relies upon a steering method, *i.e.* an algorithm that computes paths without taking into account the obstacles of the environment, which is then embedded within a general path planning scheme in order to deal with the obstacles and thus solve the full problem. The paths computed are made up of line segments, circular arcs and clothoid arcs.

Keywords — mobile-robot, path-planning, non-holonomic-system

Acknowledgements — this work was partially supported by the French programme "La Route Automatise": `<http://www-lara.inria.fr/>`.

[a] Inst. Nat. de Recherche en Informatique et en Automatique.

[b] Lab. d'Informatique GRAphique, VIsion et Robotique de Grenoble.

From Reeds and Shepp's to Continuous-Curvature Paths[*]

Th. Fraichard, A. Scheuer and R. Desvigne
Inria Rhne-Alpes & Gravir
Zirst. 655 av. de l'Europe. 38330 Montbonnot Saint Martin. France
thierry.fraichard@inria.fr

Abstract

Most path planners for car-like robots compute "Reeds and Shepp paths" made up of line segments connected with circular arcs. Such paths have a discontinuous curvature that makes them difficult to track (curvature is related to the orientation of the front wheels). The purpose of this paper is to present one of the first path planner for car-like robots that computes paths with continuous-curvature and upper-bounded curvature derivative (curvature derivative is related to the steering velocity). The approach presented herein relies upon a steering method, i.e. *an algorithm that computes paths without taking into account the obstacles of the environment, which is then embedded within a general path planning scheme in order to deal with the obstacles and thus solve the full problem. The paths computed are made up of line segments, circular arcs and clothoid arcs.*

1 Introduction

Ever since Laumond's pioneering paper in 1986 [8], a lot of research works have addressed path planning for nonholonomic systems in general and car-like robots in particular (the reader is referred to [10] for an up-to-date review on this topic). Nonholonomic systems are subject to kinematic constraints that restrict their admissible directions of motion. Car-like robots are archetypal nonholonomic systems: they can only move forward or backward in a direction perpendicular to the orientation of their rear wheels axle; besides their turning radius is lower bounded (because of the mechanical limits on the steering angle).

The study of the research works that plan collision-free paths for car-like robots shows that most of them compute paths made up of line segments connected with tangential circular arcs of minimum radius, *e.g.* [1, 5, 7, 9, 17]. One reason for this choice is that it has been shown by Reeds and Shepp that the shortest path between two configurations of a car-like robot is such a path [12]. No doubt that another reason for this choice is that they are easy to deal with from a computational point of view.

However the curvature of this type of path is discontinuous: discontinuities occur at the transitions between segments and arcs. The curvature is directly related to the orientation of the front wheels of the car. Accordingly, if a car were to track precisely such a type of path, it would have to stop at each curvature discontinuity so as to reorient its front wheels. It is therefore desirable to plan continuous-curvature paths[1]. Besides, since the derivative of the curvature is directly related to the steering velocity of the car, it is also desirable that the derivative of the curvature be upper-bounded.

The purpose of this paper is therefore to address the problem of planning collision-free paths with continuous-curvature and upper-bounded curvature derivative for car-like robots. Solutions that compute paths with continuous-curvature and upper-bounded curvature derivative have been proposed before but they never seemed to address the full problem: for instance, the lower bound on the turning radius or the obstacles would be disregarded, *e.g.* [2]. This paper is an attempt to bridge this gap. In [14], the authors addressed the case of a car moving forward only. The work reported here is the extension to the case of a car moving both forward and backward.

The approach chosen to solve the path planning problem at hand relies upon the design of a steering method[2], *i.e.* an algorithm that computes a path between two configurations without taking into account the obstacles of the environment. Given such a steering method, it is possible to use it within a general path planning scheme such as the Probabilistic Path Planner [17] or the Ariadne's Clew Algorithm [11] or the Holonomic Path Approximation Algorithm [10], in order to solve the full problem.

The steering method is therefore a key component in this planning scheme and the main contribution of this paper is the first steering method that computes paths with continuous-curvature and upper-bounded curvature derivative for car-like robots. The paper is organized as follows: the problem at hand is stated in §2, it introduces a model for car-like robots. whose properties (controllability, optimal paths, etc.) are explored in §3. The steering method proposed is then described in §4 while §5 presents experimental results.

2 Statement of the Problem

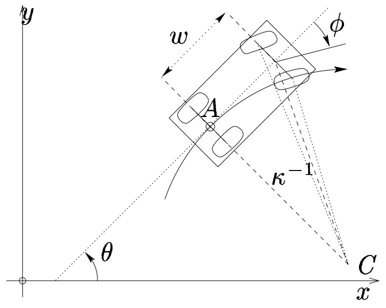

Fig. 1: a car-like robot.

[*] This work was partially supported by the French programme "La Route Automatise": <http://www-lara.inria.fr/>.

[1] As a matter of fact, it is emphasized in [3] that feedback controllers for car-like robots require this property in order to guarantee the exact reproducibility of a path.

[2] 'Steering method' is a term borrowed from [10].

Model of a Car-Like Robot. Let $\mathcal{A}$ be a car-like robot, it moves on a planar workspace $\mathcal{W} \equiv \mathbb{R}^2$ cluttered up with a set of stationary obstacles $\mathcal{B}_i, i \in \{1, \ldots, b\}$, modelled as forbidden regions of $\mathcal{W}$. $\mathcal{A}$ is modelled as a rigid body moving on the plane supported by four wheels making point contact with the ground: two rear wheels and two directional front wheels. It takes three parametres to characterize the position and orientation of $\mathcal{A}$ and an additional parametre to characterize the orientation of its front wheels. As per [2], a *configuration* of $\mathcal{A}$ is defined by the 4-tuple $q = (x, y, \theta, \kappa) \in \mathbb{R}^2 \times S^1 \times \mathbb{R}$ where (x, y) are the coordinates of the rear axle midpoint A and θ the orientation of $\mathcal{A}$ (Fig. 1). κ is the curvature of the xy-curve traced by A, it is used to characterize the orientation of the front wheels of $\mathcal{A}$: $\kappa = w^{-1} \tan \phi$, where w is the wheelbase of $\mathcal{A}$ and ϕ its steering angle[3]. Note that considering κ as a configuration parametre ensures that it will vary continuously.

Under perfect rolling assumption, a wheel moves in a direction normal to its axle. Therefore A must move in a direction normal to the rear wheels axle and the following constraint holds accordingly (*perfect rolling constraint*):

$$\left\{ \begin{array}{lcl} \dot{x} & = & v \cos \theta \\ \dot{y} & = & v \sin \theta \end{array} \right. \tag{1}$$

where v is the *driving velocity* of A, $|v| \leq v_{\max}$ ($\mathcal{A}$ moves forward when $v > 0$, stands still when $v = 0$, and moves backward when $v < 0$). The steering angle is mechanically limited, $|\phi| \leq \phi_{\max}$, and the following constraint holds (*bounded curvature constraint*):

$$|\kappa| \leq \kappa_{\max} = w^{-1} \tan \phi_{\max} \tag{2}$$

Let σ denote the derivative of κ: $\sigma = \dot{\phi} / \cos^2 \phi$. The *steering velocity* of $\mathcal{A}$ is physically limited, $|\dot{\phi}| \leq \dot{\phi}_{\max}$, and the following constraint is introduced (*bounded curvature derivative constraint*):

$$|\sigma| \leq \sigma_{\max} = \dot{\phi}_{\max} \tag{3}$$

Accordingly the model of $\mathcal{A}$ can be described by the following differential system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \sigma \tag{4}$$

with $|\kappa| \leq \kappa_{\max}$, $|v| \leq v_{\max}$ and $|\sigma| \leq \sigma_{\max}$. Henceforth the term *CC car* (for Continuous-Curvature car) is used to denote a car-like robot whose model is (4).

Admissible Paths. Let Π denote a *path* for $\mathcal{A}$, it is a continuous sequence of configurations: $\Pi(t) = (x(t), y(t), \theta(t), \kappa(t))$. An *admissible path* is a solution of the differential system (4); it is such that:

$$\left\{ \begin{array}{lcl} x(t) & = & x(0) + \int_0^t v(\tau) \cos \theta(\tau) d\tau \\ y(t) & = & x(0) + \int_0^t v(\tau) \sin \theta(\tau) d\tau \\ \theta(t) & = & \theta(0) + \int_0^t v(\tau) \kappa(\tau) d\tau \\ \kappa(t) & = & \kappa(0) + \int_0^t \sigma(\tau) \tau \end{array} \right. \tag{5}$$

with $|\kappa(t)| \leq \kappa_{\max}$, $|v(t)| \leq v_{\max}$ and $|\sigma(t)| \leq \sigma_{\max}$. The focus in this paper is on shortest path planning, it is therefore assumed that $|v(t)| = 1$ (thus the time and the arc length of a path are the same). Then an admissible path Π is fully characterized by its start configuration q_s, its length l, its curvature profile $\kappa : [0, l] \longrightarrow [-\kappa_{\max}, \kappa_{\max}]$, and its speed profile $v : [0, l] \longrightarrow \{-1, 1\}$.

Planning Problem. Given a start configuration q_s and a goal configuration q_g, find an admissible path $\Pi = (q_s, l, \kappa, v)$ such that:

- Π links q_s and q_g, *i.e.* $\Pi(0) = q_s$ and $\Pi(l) = q_g$.
- Π is *collision-free*, *i.e.*:

  $$\forall t \in [0, l], \forall i \in \{1, \ldots, b\}, \mathcal{A}(\Pi(t)) \cap \mathcal{B}_i = \emptyset$$

  where $\mathcal{A}(q)$ denotes the region of $\mathcal{W}$ occupied by $\mathcal{A}$ at configuration q.

Finally the length of Π should be minimum.

3 Properties of the CC Car

A CC car is a small-time controllable system[4] [15, Theorem 1]. The small-time controllability of a CC car implies that the existence of an admissible collision-free path is equivalent to the existence of a collision-free path [10, Theorem 3.1]. As far as optimal —*i.e.* shortest— paths are concerned it can be shown that, in the absence of obstacles, if a path exists between two configurations then an optimal path exists [15, Theorem 2]. The nature of the optimal paths is more difficult to establish. However [13] demonstrates that, for the forward CC car, *i.e.* the CC car moving forward only, the optimal paths are made up of: (a) line segments, (b) circular arcs of radius $1/\kappa_{\max}$, and (c) clothoid arcs[5] of sharpness $\pm\sigma_{\max}$. Unfortunately, it appears that, whenever the shortest path includes a line segment, it is irregular and contains an infinite number of clothoid arcs that accumulate towards each endpoint of the segment [2]. Furthermore, when the distance between two configurations is large enough, the shortest path contains a line segment (hence an infinite number of clothoid arcs) [4].

In summary, although the exact nature of the optimal paths for the CC car has not been established yet, it seems reasonable to conjecture that they will (at least) be made up of line segments, circular arcs and clothoid arcs, and that they will be irregular in most cases.

4 Steering the CC Car

As mentioned in §1, the approach chosen to solve the path planning problem at hand relies upon the design of a steering method, *i.e.* an algorithm that computes an admissible path between two configurations without taking into account the obstacles of the environment. The steering method computes *steering paths*.

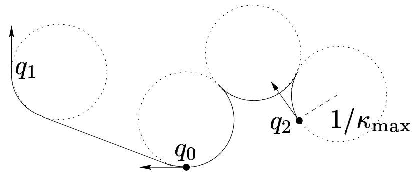

Fig. 2: computing Reeds and Shepp's paths.

In the absence of obstacle, optimal paths are the natural choice for the steering paths. Unfortunately, the conjectured irregularity of the optimal paths for the CC car prevents their use (*cf.* §3). It was decided instead to compute steering paths derived from the "Reeds and Shepp's paths" (henceforth called RS paths) [12]. The RS path between two configurations is made

[3] ϕ is the average orientation of the two front wheels of $\mathcal{A}$.

[4] The set of configurations reachable from q before a time t contains a neighbourhood of q for any t.

[5] A clothoid is a curve whose curvature varies linearly with its arc length.

up of line segments and circular arcs of radius $1/\kappa_{\max}$. The circular arcs are supported by the circles of radius $1/\kappa_{\max}$ tangent to the configurations considered or tangent to two of these circles. The line segments are tangent to these circles (Fig. 2). Note the key role played by the circles of radius $1/\kappa_{\max}$ tangent to the start and goal configurations. These circles represent the locus of the set of configurations that can be reached by moving forward/backward while turning to the left/right with the maximum steering angle.

Steering paths are similar to RS paths but, in order to ensure curvature continuity, the circular arcs are replaced by transitions called *CC turns* whose curvature varies continuously from 0 up and then down back to 0, and that are made up of circular arcs of radius $1/\kappa_{\max}$ and clothoid arcs.

As explained further down, CC turns are defined so that they yield circles similar to the tangent circles used to compute RS paths. Accordingly CC turns and line segments can be combined in order to form the steering paths in the way circular arcs and line segments are combined to form the RS paths. Steering paths are made up of a finite number of (a) line segments, (b) circular arcs of radius $1/\kappa_{\max}$, and (c) clothoid arcs. They are not optimal but, based upon the result already established in [15] for the forward CC car, it is conjectured that they are suboptimal, *i.e.* longer than the optimal path of no more than a given constant. This result is yet to be demonstrated however.

The next two sections respectively presents the CC turns and their properties (§4.1), and the way to compute steering paths (§4.2).

4.1 CC Turns

A natural way to change the car orientation is to: (a) move forward while turning the front wheels as fast as possible. (b) follow a circular arc once the maximum steering angle is reached, and (c) move forward while turning the front wheels back to a null steering angle. Hence the following definition of a CC turn: in the general case, a CC turn is made up of three parts: (a) a clothoid arc of sharpness $\sigma = \pm\sigma_{\max}$ whose curvature varies from 0 to $\kappa_{\max}$, (b) a circular arc of radius $1/\kappa_{\max}$, and (c) a clothoid arc of sharpness $-\sigma$ whose curvature varies from $\kappa_{\max}$ to 0. CC turns were first introduced in [14] for the forward CC car. The key properties of CC turns are recalled and an extension for the CC car is proposed afterwards.

4.1.1 CC Turns for the Forward CC Car

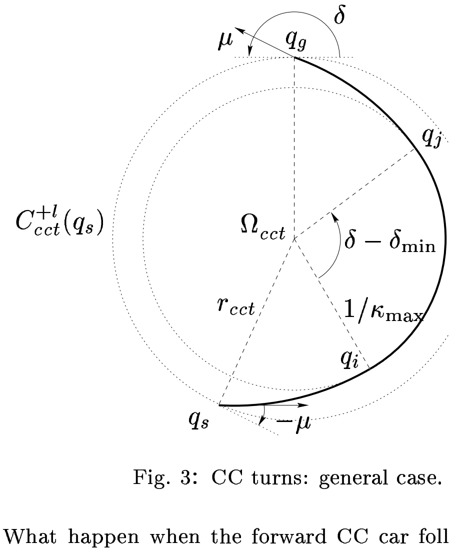

Fig. 3: CC turns: general case.

What happen when the forward CC car follows a CC turn is illustrated in Fig. 3. Let $q_s = (x_s, y_s, \theta_s, 0)$ be the start configuration. It is assumed that the forward CC car moves forward while turning to the left. First it follows a clothoid arc of sharpness $\sigma_{\max}$ until it reaches $q_i = (x_i, y_i, \theta_i, \kappa_{\max})$. Then it follows a circular arc of radius $1/\kappa_{\max}$ until it reaches $q_j = (x_j, y_j, \theta_j, \kappa_{\max})$. The centre of this circular arc is:

$$\Omega_{cct} = \left\{ \begin{array}{l} x_{cct} = x_i - \sin\theta_i/\kappa_{\max} \\ y_{cct} = y_i + \cos\theta_i/\kappa_{\max} \end{array} \right.$$

Finally it follows a clothoid arc of sharpness $-\sigma_{\max}$ until it reaches the goal configuration $q_g = (x_g, y_g, \theta_g, 0)$. Let $\delta = \theta_g - \theta_s$ denote the change of orientation between q_s and q. δ is the *deflection* of the CC turn, it is used to characterize CC turns; $\delta \in [0, 2\pi[$ for left CC turns.

It is the measure of the circular arc of a CC turn that actually determines where the goal configuration is and the main result established in [14] is that the locus of the goal configurations is a circle $C^{+l}_{cct}(q_s)$ whose centre is Ω_{cct} and whose radius is $r_{cct} = \sqrt{(x_{cct} - x_s)^2 + (y_{cct} - y_s)^2}$. In addition, the angle μ between the orientation of q_g and the tangent to $C^{+l}_{cct}(q_s)$ at q_g is constant; it is the opposite of the angle between the orientation of q_s and the tangent to $C^{+l}_{cct}(q_s)$ at q_s (Fig. 3).

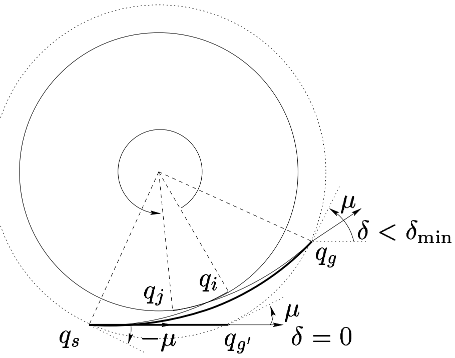

Fig. 4: CC turns: "$\delta = 0$" and "$0 < \delta < \delta_{\min}$" cases.

Let $\delta_{\min} = {\kappa_{\max}}^2/\sigma_{\max}$ be the deflection of the CC turn whose circular arc has zero length. With the above definition, a CC turn of deflection $0 < \delta < \delta_{\min}$ makes a loop and intersects itself (Fig. 4). In this case, it is proposed in [14] to use instead a loopless and shorter path made up of: (a) a clothoid arc of sharpness $\sigma \leq \sigma_{\max}$ and (b) a symmetric clothoid arc of sharpness $-\sigma$. Such a path is admissible and it is shown in [14] that there is a unique σ such that the goal configuration belongs to $C^{+l}_{cct}(q_s)$.

As for the $\delta = 0$ case, the CC turn becomes the line segment of length $2r_{cct}\sin\mu$ so as to ensure that the goal configuration also belongs to $C^{+l}_{cct}(q_s)$ (Fig. 4).

4.1.2 CC Turns for the CC Car

CC turns allow the forward CC car to reach any goal configuration q_g with a null curvature which is located on the circle $C^{+l}_{cct}(q_s)$ and such that the angle between the orientation of q_g and the tangent to $C^{+l}_{cct}(q_s)$ at q_g is constant and equal to the opposite of the angle between the orientation of q_s and the tangent to $C^{+l}_{cct}(q_s)$ at q_s. Extending the CC turns to CC cars consists in taking advantage of the fact that the CC car can make back and forth motion so as to produce shorter CC turns while keeping the above property satisfied.

A straightforward extension concerns the CC turns of large deflection for which it is shorter, once q_i is reached, to back up to q_j instead of moving forward (Fig. 5). The limit deflection for which it is shorter to back up is determined as follows: the measure of the circular arc of the CC turn of deflection δ is $\delta - \delta_{\min}$ when the CC car moves forward from q_i to q_j, and

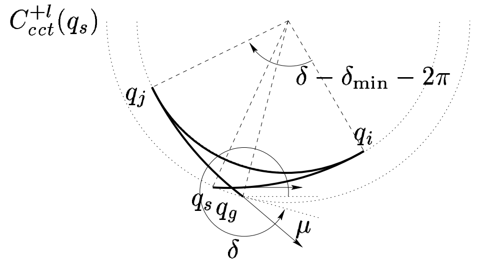

Fig. 5: CC turns: "$\delta_{\min}+\pi \le \delta < 2\pi$" case.

$\delta-\delta_{\min}-2\pi$ when it moves backward. Accordingly the backward motion is shorter when $\delta \ge \delta_{\min}+\pi$.

In summary, the CC turns as defined above allow the CC car to reach any goal configuration q_g with a null curvature which is located on the circle $C^{+l}_{cct}(q_s)$ and such that the angle between the orientation of q_g and the tangent to $C^{+l}_{cct}(q_s)$ at q_g is constant and equal to the opposite of the angle between the orientation of q_s and the tangent to $C^{+l}_{cct}(q_s)$ at q_s. It is the deflection associated with q_g that determines the nature of the CC turn:

- $\boldsymbol{\delta = 0}$: the CC turn is a line segment.
- $\boldsymbol{0 < \delta < \delta_{\min}}$: the CC turn is made up of (a) a clothoid arc of sharpness $\sigma \le \sigma_{\max}$ and (b) a symmetric clothoid arc of sharpness $-\sigma$.
- $\boldsymbol{\delta_{\min} \le \delta < \delta_{\min}+\pi}$: the CC turn is made up of (a) a clothoid arc of sharpness $\sigma_{\max}$ whose curvature varies from 0 to $\kappa_{\max}$, (b) a *forward* circular arc of radius $1/\kappa_{\max}$, and (c) a clothoid arc of sharpness $-\sigma_{\max}$ whose curvature varies from $\kappa_{\max}$ to 0.
- $\boldsymbol{\delta_{\min}+\pi \le \delta < 2\pi}$: the CC turn is made up of (a) a clothoid arc of sharpness $\sigma_{\max}$ whose curvature varies from 0 to $\kappa_{\max}$, (b) a *backward* circular arc of radius $1/\kappa_{\max}$, and (c) a clothoid arc of sharpness $-\sigma_{\max}$ whose curvature varies from $\kappa_{\max}$ to 0.

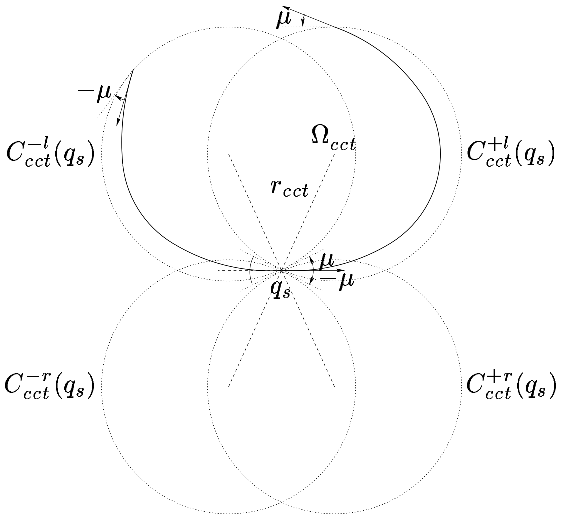

Fig. 6: $C^{+l}_{cct}(q_s)$, $C^{+r}_{cct}(q_s)$, $C^{-l}_{cct}(q_s)$ and $C^{-r}_{cct}(q_s)$.

The above analysis was carried out for the case of the CC car moving forward while turning to the left. The case where the CC car is turning to the right is dealt with in the same manner, it yields a symmetric circle $C^{+r}_{cct}(q_s)$ (Fig. 6). Two similar circles, $C^{-l}_{cct}(q_s)$ and $C^{-r}_{cct}(q_s)$, represent the locus of the set of configurations with null curvature allowing the CC car to reach q_s with a CC turn turning to the left or to the right (Fig. 6). They are readily obtained by considering that the CC car moves backward (*cf.* [13]).

4.1.3 Arc Length of the CC turns

The arc length of a CC turn depends upon its nature which in turn depends upon its deflection δ. Let $l(\delta)$ denote the arc length of a CC turn of deflection δ, it is defined as:

- $\boldsymbol{\delta = 0}$: the CC turn is a line segment of length $l(0) = 2r_{cct}\sin\mu$.
- $\boldsymbol{0 < \delta < \delta_{\min}}$: let σ be the sharpness characterizing the CC turn in this case: $l(\delta) = 2\sqrt{\delta/\sigma}$. $l(\delta)$ increases monotonously from $2r_{cct}\sin\mu$ to $l_{\min} = 2\kappa_{\max}/\sigma_{\max}$. $l_{\min}$ is the arc length of the general CC turn whose circular arc has zero length.
- $\boldsymbol{\delta_{\min} \le \delta < \delta_{\min}+\pi}$: in this case, the arc length of the CC turn is $l_{\min}$ plus the arc length of its circular arc: $l(\delta) = l_{\min} + (\delta-\delta_{\min})/\kappa_{\max}$. $l(\delta)$ increases linearly from $l_{\min}$ to $l_{\min}+\pi/\kappa_{\max}$.
- $\boldsymbol{\delta_{\min}+\pi \le \delta < 2\pi}$: this is the same case as above: $l(\delta) = l_{\min} + (2\pi-\delta+\delta_{\min})/\kappa_{\max}$. $l(\delta)$ decreases linearly from $l_{\min}+\pi/\kappa_{\max}$ to $l_{\min}+\delta_{\min}/\kappa_{\max} = 3\kappa_{\max}/\sigma_{\max}$.

4.2 Steering Paths

Let CC-steering denote the function that computes the steering paths for the CC car. As mentioned in §4, it is derived from the function that computes RS paths. The RS path between two configurations is the shortest among a set of paths that belong to one of the nine following families [16]:

$$\begin{array}{ll} (i) & l^+l^-l^+ \text{ or } r^+r^-r^+ \\ (ii)(iii) & A|AA \text{ or } AA|A \\ (iv) & AA|AA \\ (v) & A|AA|A \\ (vi) & A|ASA|A \\ (vii)(viii) & A|ASA \text{ or } ASA|A \\ (ix) & ASA \end{array} \tag{6}$$

where A (resp. S) denotes a circular arc (resp. line segment). $|$ denotes a change of direction of motion (a cusp point). A may be replaced by r or l to specify a right (clockwise) or left (counterclockwise) turn. A + or - superscript indicates a forward or backward motion. CC-steering considers the same nine families of paths but the circular arcs are replaced by CC turns. Like RS paths, the steering paths are determined through the analysis of the tangency relationships that exists between the circles C^{+l}_{cct}, C^{+r}_{cct}, C^{-l}_{cct} and C^{-r}_{cct} defined earlier and attached to the start and goal configurations.

Noting that the ASA, AA and $A|A$ paths are building blocks for the steering paths of the nine families above, the condition of existence and the characteristics of these three types of path are studied first in §4.2.1, §4.2.2 and §4.2.3. Then CC steering is presented in §4.2.4.

4.2.1 *ASA* Paths

A RS path of type ASA is obtained by searching the tangent (internal or external) to two of the circles of radius $1/\kappa_{\max}$ associated with the start and goal configurations. The steering paths of type ASA are obtained in a similar way by using the circles C^{+l}_{cct}, C^{+r}_{cct}, C^{-l}_{cct} and C^{-r}_{cct} defined earlier. However, due to the fact that the orientations of the configurations located on these circles make a constant angle μ with the tangent to these circles, the connecting line is not a tangent to the circles but rather a μ-tangent: the μ-tangent crosses the two circles and in both cases makes an angle μ with the tangent at the intersection points. Fig. 7 illustrates how μ-tangents are obtained. C^1_{cct}

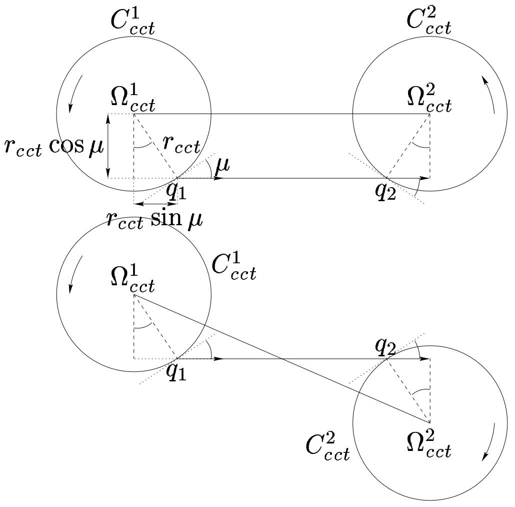

Fig. 7: *ASA* paths.

and C^2_{cct} are the two circles associated with the start and goal configurations. An inside arrow indicates the corresponding direction of motion (clockwise or counterclockwise). Two cases must be considered:

- C^1_{cct} and C^2_{cct} have the same direction of motion: the μ-tangent is external and parallel to the line of centres $\Omega^1_{cct}\Omega^2_{cct}$. A straightforward geometric analysis shows that an external μ-tangent always exists but that the line segment q_1q_2 exists iff $\Omega^1_{cct}\Omega^2_{cct} \geq 2r_{cct}\sin\mu$ (Fig. 7 top). The length of the line segment q_1q_2 is:

$$l(q_1q_2) = \Omega^1_{cct}\Omega^2_{cct} - 2r_{cct}\sin\mu \qquad (7)$$

- C^1_{cct} and C^2_{cct} have an opposite direction of motion: the μ-tangent is internal and crosses the line of centres $\Omega^1_{cct}\Omega^2_{cct}$. An internal μ-tangent exists iff $\Omega^1_{cct}\Omega^2_{cct} \geq 2r_{cct}\cos\mu$ but the line segment q_1q_2 exists iff $\Omega^1_{cct}\Omega^2_{cct} \geq 2r_{cct}$ (Fig. 7 bottom). The length of the line segment q_1q_2 is (*cf.* [13]):

$$l(q_1q_2) = \frac{{\Omega^1_{cct}\Omega^2_{cct}}^2 - 4{r_{cct}}^2}{\sqrt{{\Omega^1_{cct}\Omega^2_{cct}}^2 - 4{r_{cct}}^2\cos^2\mu} + 2r_{cct}\sin\mu} \qquad (8)$$

4.2.2 *AA* Paths

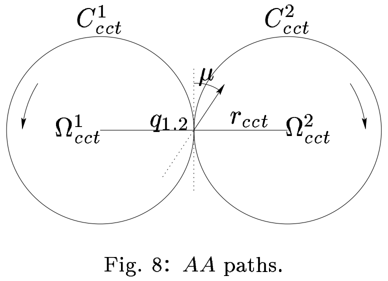

Fig. 8: *AA* paths.

Fig. 8 illustrates how a steering path of type *AA* is obtained. It involves two circles C^1_{cct} and C^2_{cct} with an opposite direction of motion. Let $q_{1.2}$ denote the point of contact between the two circles C^1_{cct} and C^2_{cct}. It belongs to the line of centres $\Omega^1_{cct}\Omega^2_{cct}$. Accordingly such a path exists iff $\Omega^1_{cct}\Omega^2_{cct} = 2r_{cct}$.

4.2.3 *A|A* Paths

Fig. 9 illustrates how a steering path of type $A|A$ is obtained. In this case, it involves two circles C^1_{cct} and C^2_{cct} with the same direction of motion and $q_{1.2}$ does not belong to the line of centres $\Omega^1_{cct}\Omega^2_{cct}$. It can be shown that $\Omega^1_{cct}\Omega^2_{cct} = 2r_{cct}\cos\mu$; it is the condition of existence of such a path.

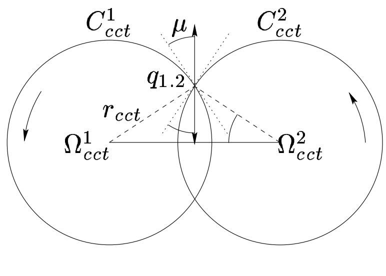

Fig. 9: $A|A$ paths.

4.2.4 CC-Steering

It is straightforward to combine the conditions of existence of the *ASA*, *AA* and $A|A$ paths in order to determine the condition of existence of a steering path belonging to one of the nine families presented earlier. Once the existence of a given type of steering path is assessed, it is also straightforward to compute it and determine its arc length using (7), (8) and the results established in §4.1.3.

Given a start and a goal configurations q_s and q_g, CC-steering determines for each family of steering paths and for each possible pair of circles C^{+l}_{cct}, C^{+r}_{cct}, C^{-l}_{cct} and C^{-r}_{cct} whether the corresponding steering path exists. If so, CC steering computes its length. Then the shortest of these paths is selected as being the steering path between q_s and q_g.

The conditions of existence of the steering paths of the different families guarantee that at least one such path exists. Accordingly CC steering is complete, it always return a steering path linking q_s and q_g.

5 Experimental Results

As mentioned in §1, the motivation behind the design of CC-steering was to use it within a general path planning scheme so as to solve the full path planning problem as stated in §2. The Probabilistic Path Planner [17] has been selected to be the general path planning scheme because it is efficient in practice and easy to implement. It is a two stage algorithm: in the first stage, it builds a graph that captures the connectivity of the collision-free space of the robot. The nodes of the graph are collision-free configurations picked up at random. The edges of the graph are collision-free steering paths computed thanks to CC-steering and a collision-checking function such as the one proposed in [14]. In the second stage, the graph is used to solve specific path planning problems between a start and a goal configurations.

	min.	average	max.	deviation
ratio	1.00253	1.1065	2.45586	0.172188

Table 1: RS *vs.* CC paths' length.

RS (1000 paths)	CC (1000 paths)	average ratio
3.466586 s.	4.483492 s.	1.33

Table 2: RS *vs.* CC paths' computation time.

CC steering has been implemented[6] along with a function computing RS paths and their results have been compared. Fig. 10 illustrates the results obtained. It appears that, for a given pair of (initial, goal) configurations, the resulting RS and CC paths may belong to the same family of path (Fig. 10, top left), or to different families. CC paths may have the same number of back up manoeuvres (Fig. 10, top right), more (Fig. 10, bottom left) or less (Fig. 10, bottom right).

[6] In C^{++} on a Pentium II 400 MHz. bi-processor PC.

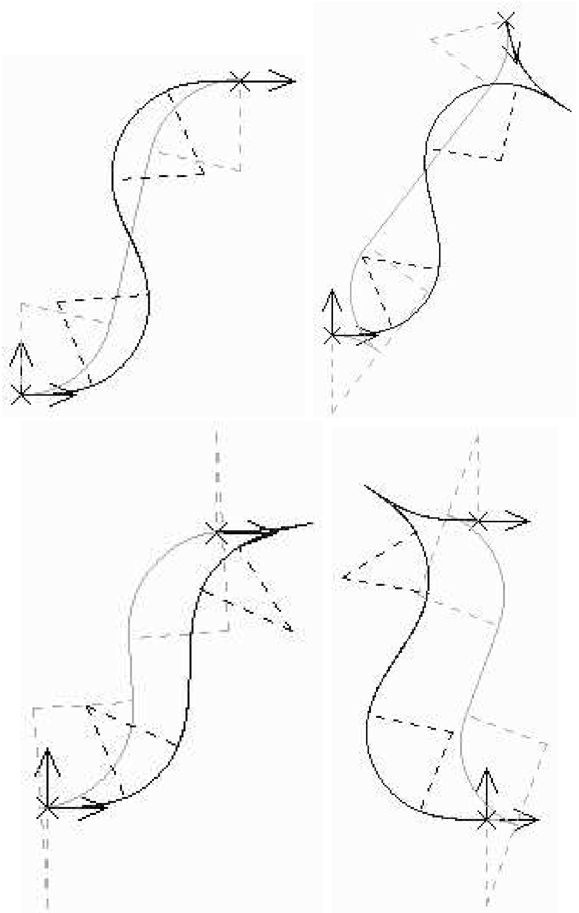

Fig. 10: paths planning results for the CC car.

Further comparisons were made regarding the respective length of the paths and the time required for their computation. The ratio of CC over RS paths' lengths were computed for one hundred pairs of (initial, goal) configurations. The results obtained are summarized in Table 1. In most cases (82%), CC paths are only about 10% longer than RS paths. Similar experiments were carried out for the computation time. The running time of both RS and CC-steering are of the same order of magnitude (Table 2). Given that continuous curvature paths can be tracked with a much greater accuracy by a real car-like vehicle (*cf.* the experimental results obtained in [15]), the results reported herein demonstrate the interest of CC paths (about the same computation time and same length).

The integration of CC steering within the Probabilistic Path Planning scheme is straightforward. Due to lack of space, the results obtained are not reported here, the reader is referred to [6] for more details.

References

[1] J. Barraquand and J.-C. Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2):77–103, 1989.

[2] J.-D. Boissonnat, A. Crzo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Research Report 2160, Inst. Nat. de Recherche en Informatique et en Automatique, Rocquencourt (FR), January 1994.

[3] A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 171–253. Springer, 1998.

[4] E. Degtiariova-Kostova and V. Kostov. Irregularity of optimal trajectories in a control problem for a car-like robot. Research Report 3411, Inst. Nat. de Recherche en Informatique et en Automatique, April 1998.

[5] Th. Fraichard. Smooth trajectory planning for a car in a structured world. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 318–323, Sacramento, CA (US), April 1991.

[6] Th. Fraichard, A. Scheuer, and R. Desvigne. From reeds and shepp's to continuous-curvature paths. Research Report, Inst. Nat. de Recherche en Informatique et en Automatique, 1999. `<http://www.inria.fr/RRRT/publications-eng.html>`.

[7] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2–7, Scottsdale, AZ (US), May 1989.

[8] J.-P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Proc. of the Int. Conf. on Intelligent Autonomous Systems*, pages 346–354, Amsterdam (NL), December 1986.

[9] J.-P. Laumond, P. E. Jacobs, M. Tax, and R. M. Murray. A motion planner for non-holonomic mobile robots. *IEEE Trans. Robotics and Automation*, 10(5):577–593, October 1994.

[10] J.-P. Laumond, S. Sekhavat, and F. Lamiraux. Guidelines in nonholonomic motion planning for mobile robots. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 1–53. Springer, 1998.

[11] E. Mazer, P. Bessire, and J.-M. Ahuactzin. The ariadne's clew algorithm. *Journ. of Artificial Intelligence Research*, 9:295–316, July-December 1998.

[12] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[13] A. Scheuer. *Planification de chemins courbure continue pour robot mobile non-holonome.* Thse de doctorat, Inst. Nat. Polytechnique de Grenoble, Grenoble (FR), January 1998.

[14] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 997–1003, Grenoble (FR), September 1997.

[15] A. Scheuer and Ch. Laugier. Planning sub-optimal and continuous-curvature paths for car-like robots. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 25–31, Victoria, BC (CA), October 1998.

[16] P. Soures and J.D. Boissonnat. Optimal trajectories for nonholonomic mobile robots. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 93–170. Springer, 1998.

[17] P. Svestka and M. H. Overmars. Probabilistic path planning. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *Lecture Notes in Control and Information Science*, pages 255–304. Springer, 1998.