Continuous-Curvature Path Planning for Car-Like Vehicles

A. Scheuer and Th. Fraichard

Inria* Rhône-Alpes & Gravir†

Zirst. 655 avenue de l'Europe. 38330 Montbonnot Saint Martin. France

[Alexis.Scheuer, Thierry.Fraichard]@inria.fr

Abstract

In this paper, we consider path planning for a car-like vehicle. Previous solutions to this problem computed paths made up of circular arcs connected by tangential line segments. Such paths have a non continuous curvature profile. Accordingly a vehicle following such a path has to stop at each curvature discontinuity in order to reorient its front wheels. To remove this limitation, we add a continuous-curvature constraint to the problem at hand. In addition, we introduce a constraint on the curvature derivative, so as to reflect the fact that a car-like vehicle can only reorient its front wheels with a finite velocity.

We propose an efficient solution to the problem at hand that relies upon the definition of a set of paths with continuous curvature and maximum curvature derivative. These paths contain at most eight pieces, each piece being either a line segment, a circular arc of maximum curvature, or a clothoid arc. They are called SCC-paths *(for Simple Continuous Curvature paths). They are used to design a* local path planner, *i.e. a non-complete collision-free path planner, which in turn is embedded in a global path planning scheme. The result is the first path planner for a car-like vehicle that generates collision-free paths with continuous curvature and maximum curvature derivative. Experimental results are presented.*

1 Introduction

In this paper, we focus on the path planning problem for a car-like vehicle that goes only forward. Such a vehicle is subject to two non-holonomic constraints: it can only move along a direction perpendicular to its rear wheels axle (continuous tangent direction), and its turning radius is lower bounded (maximum curvature) [1]. Numerous works, e.g. [1, 14, 12, 20], have been done to plan paths for such vehicles, but almost all of them generate sequence of Dubins' curves [5], i.e. paths made of circular arcs connected by tangential line segments. The main reason for this is that these paths are the shortest ones for such a vehicle [5]. The main drawback of these paths is that their curvature is not continuous. Accordingly a vehicle following such a path has to stop at each curvature discontinuity in order to reorient its front wheels.

Since we are interested in planning forward paths only, i.e. paths without manoeuvres, we do not want the vehicle to stop, except possibly at the initial and final configurations. For this reason, we add a continuous-curvature constraint to the classical non-holonomic path planning problem for car-like vehicles. In addition, we introduce a constraint on the curvature derivative; it is upper bounded so as to reflect the fact that the vehicle can only reorient its front wheels with a finite velocity.

Addressing a similar problem (but without the maximum curvature constraint), Boissonnat et al. [2] proved, using the Pontryagin's Maximum Principle, that the shortest path between two vehicle's configurations is made up of line segments and clothoid[1] arcs of maximum curvature derivative. Later, Kostov and Degtiariova-Kostova proved that these shortest paths are, in the general case, made of an infinity of pieces [10, 11].

Similar results can be extended to the particular problem we consider, adding circular arcs of maximum curvature to the set of locally optimal paths. Therefore, in order to come up with a practical solution to the problem at hand, we define a set of paths, derived from Dubins' curves, that have continuous curvature and maximum curvature derivative. These paths contain at most eight pieces, each piece being either a line segment, a circular arc of maximum curvature, or a clothoid arc. They are called *SCC-paths* (for Simple Continuous Curvature paths). They are used to design a *local path planner*, i.e. a non-complete collision-free path planner, which in turn is embedded in a global path planning scheme, namely the Probabilistic Path Planner [21]. The result is the first path planner for a car-like vehicle that generates collision-free paths with continuous curvature and maximum curvature derivative.

After a short review of the works related to this topic (§2), the problem at hand will be stated (§3). Then the SCC-paths will be described, along with the local path planner (§4). Finally the global planner will be presented, along with experimental results (§5).

2 Related Works

As mentioned earlier, almost all the path planners for car-like vehicles return a sequence of Dubins' curves, i.e. paths made up of circular arcs connected by tangential line seg-

*Inst. Nat. de Recherche en Informatique et en Automatique.

†Lab. d'Informatique GRAphique, VIsion et Robotique de Grenoble.

[1] A clothoid is a curve whose curvature is a linear function of its arc length.

ments [1, 14, 12, 20]. Fleury et al. [6] does propose a path planner generating continuous-curvature paths, but unfortunately, it does not take into account the maximum curvature constraint and therefore do not apply to car-like vehicles.

On the other hand, several works deal with continuous-curvature path generation, i.e. computation of a path without considering the obstacle avoidance. They return curves as different as B-splines [9], quintic polynomials [19] or polar splines [15]. However it appears that the most popular curves are by far curves whose curvature is a polynomial function of their arc length, such as clothoids [8], cubic spirals [7] or, more generally, intrinsic splines [4].

Drawing upon Kanayama and Hartman's work [7], we previously proposed a path planner returning paths made up of pairs of clothoid arcs [17, 18]. To the best of our knowledge, it was the first continuous curvature path planner for a car-like vehicle. However it did not take into account the maximum curvature derivative constraint. Besides, because the path generated contained only clothoid arcs, they were sometimes much longer than needed. Introducing line segments and circular arcs, that are locally optimal, is a step toward sub-optimal paths.

3 Statement of the Problem

First the model of the car-like vehicle and its workspace are presented. Then the feasible paths for such a vehicle are described. Finally the path planning problem considered is stated and our approach to solve it is sketched.

3.1 A Car-like Vehicle

The model of our robot $\mathcal{A}$ represents a car-like vehicle in a planar environment (Fig. 1). It is a rigid body moving on a plane, supported by four wheels making point contact with the ground. The two front wheels are directional, and the rear wheels axle is fixed with respect to $\mathcal{A}$'s body. Let R denote the reference point of $\mathcal{A}$; it is the midpoint of the rear wheels axle. A configuration of $\mathcal{A}$ is defined by the 4-tuple (x, y, θ, κ), where x and y are the coordinates of R, θ is the orientation of $\mathcal{A}$ (i.e. the angle between the x axis and the main axle of $\mathcal{A}$) and κ the inverse of the turning radius of $\mathcal{A}$ (it is defined by the orientation of its front wheels). As we will see further down, κ allows a very simple definition of what continuous curvature paths are. It was therefore selected as a configuration parameter rather than the front wheels' orientation.

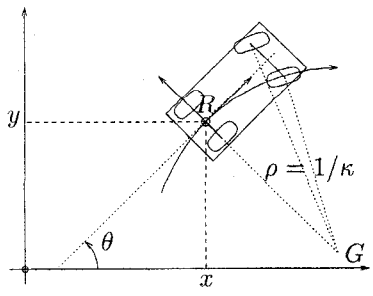

Figure 1: a car-like vehicle.

$\mathcal{A}$ moves on a planar workspace $\mathcal{W} \subset \mathbb{R}^2$, which is cluttered with a set of stationary obstacles $\mathcal{B}_j$, $j \in \{1, \ldots, b\}$. These obstacles are represented by polygonal regions of $\mathcal{W}$.

The motion of $\mathcal{A}$ is limited by two classical constraints. First, as the rear wheels axle is fixed and as the wheels should roll without sliding, the center of rotation G of the body of $\mathcal{A}$ must be located on the rear wheels axle (possibly at an infinite distance). Thus the following constraint holds:

$$-\dot{x} \sin\theta + \dot{y} \cos\theta = 0 \tag{1}$$

Second, since the orientation of the front wheels is mechanically limited, the distance ρ between R and G, i.e. the turning radius of $\mathcal{A}$, is lower bounded by a given value ρ_{min}. Accordingly the curvature κ, is upper bounded by $\kappa_{max} = 1/\rho_{min}$:

$$|\kappa| \leq \kappa_{max} \tag{2}$$

In addition the curvature derivative is upper bounded so as to reflect the fact that $\mathcal{A}$ can only reorient its front wheels with a finite velocity:

$$|\dot{\kappa}| \leq \sigma_{max} \tag{3}$$

3.2 Feasible Paths

Let $\mathcal{C}$ be $\mathcal{A}$'s configuration space: $\mathcal{C} \subset \mathcal{W} \times \mathcal{S}^1 \times [-\kappa_{max}, \kappa_{max}]$. A path is a continuous oriented curve of $\mathcal{C}$. Using constraint (1), such a path can be represented by its projection in $\mathcal{W}$. In the following, a path of $\mathcal{A}$ will be represented also by the continuous oriented curve followed by R in $\mathcal{W}$. A configuration contains the instantaneous curvature of the path followed by $\mathcal{A}$. Therefore, as a path is a continuous sequence of configurations, it has a continuous-curvature profile.

A path is *feasible* if and only if a) it is an oriented curve of $\mathcal{W}$ which is continuous, whose derivative is continuous almost everywhere with opposite semi-tangents at the cusp points, and whose second derivative is continuous between two successive cusp points, and b) its curvature and the derivative of its curvature w.r.t. the arc length remain bounded respectively by κ_{max} and σ_{max}. We suppose that the curvature can change at the cusp points. The bound σ_{max} of the derivative of the curvature w.r.t. the arc length is related to the maximum turning velocity of the steering wheel of the vehicle, when this vehicle moves at constant speed.

In addition, such a path is said to be *smooth* (or without manoeuvre, that is without any change of the motion direction) if and only if it has no cusp points, i.e. if it is C^2. A smooth path Π of finite length can be represented by its starting configuration $q(0)$, its length l and its curvature profile $\kappa : [0, l] \longmapsto [-\kappa_{max}, \kappa_{max}]$ (with $|\dot{\kappa}| \leq \sigma_{max}$).

3.3 The Path Planning Problem

A path Π is a mapping from $\mathbb{R}$ to $\mathcal{C}$: $\Pi : s \mapsto \Pi(s), \forall s \in [0, l]$, where l is the length of Π. Given a start configuration $q_s = (x_s, y_s, \theta_s, \kappa_s)$ and a goal one $q_g = (x_g, y_g, \theta_g, \kappa_g)$, such a path is a solution to our problem if and only if it links q_s to q_g and is feasible, smooth and collision-free, i.e.:

1. End conditions: $\Pi(0) = q_s$ and $\Pi(l) = q_g$;

2. Π is feasible and smooth, and therefore its curvature profile is a continuous function $\kappa : [0, l] \longmapsto [-\kappa_{max}, \kappa_{max}]$, such that $|\dot{\kappa}| \leq \sigma_{max}$;
3. Π is collision-free:

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

where $\mathcal{A}(\Pi(s))$ denotes the region of $\mathcal{W}$ occupied by $\mathcal{A}$ when in the configuration $\Pi(s)$.

3.4 The Solution

Because of the expected complexity of finding the optimal solution to the problem at hand (cf. §1), we have decided to restrict our search for a solution path to a finite and discrete set of paths. They are derived from Dubins' curves [5] and have continuous curvature and maximum curvature derivative. These paths contain at most eight pieces, each piece being either a line segment, a circular arc of maximum curvature, or a clothoid arc. They are called *SCC-paths* (for Simple Continuous Curvature paths). They are used to design a *local path planner*, i.e. a non-complete collision-free path planner (see §4). The local path planner is then embedded in a global path planning scheme, namely the Probabilistic Path Planner [21], so as to obtain the first path planner for a car-like vehicle that generates collision-free paths with continuous curvature and maximum curvature derivative (see §5).

4 The Local Path Planner

4.1 Outline

Given two configurations q_a and q_b, the local planner:

1. computes the set of simple continuous-curvature (SCC-) paths linking q_a to q_b;
2. determines, in this set, the subset of collision-free SCC-paths;
3. selects, in the resulting subset, the shortest path and returns it as **the** *simple continuous-curvature path* linking q_a to q_b.

If the subset of collision-free SCC-paths is empty, the local planner fails to link q_a to q_b.

In the general case, Dubins' curves are made of three pieces, connected so as to keep a continuous tangent direction. The two extreme pieces are circular arcs and the middle one is either a line segment or a circular arc. In the continuous-curvature case, we want to avoid the discontinuity of the curvature occurring in Dubins' curves at each connection between two pieces.

In SCC-paths, the circular arcs of Dubins' curves are replaced by turns whose curvature continuously varies from zero to a given value, then back to zero. These turns start with a null curvature because they may follow a line segment, and they end also with a null curvature because they may be followed by a line segment. The continuous curvature turns are described in §4.2. Let s denote a line segment and let l or r denote a continuous-curvature turn depending on its direction (left or right). Similarly to Dubins' curves, they are at most most six simple continuous-curvature (SCC-) paths between two configurations, defined respectively as *lsl*, *lsr*, *rsl*, *rsr*, *lrl* and *rlr*.

Continuous curvature turns are presented in §4.2. Then SCC-paths are defined in §4.3. Finally §4.4 gives more details on collision checking.

4.2 Continuous Curvature Turns

As mentioned earlier, continuous curvature turns are made of three pieces:

1. a first clothoid arc of sharpness (i.e. constant derivative of the curvature w.r.t. the arc length) $\sigma = \pm\sigma_{max}$ and of length $l = \kappa_{max}/\sigma_{max}$,
2. an optional circular arc of radius $1/\kappa_{max}$, and
3. a second clothoid arc of sharpness -σ and of length l.

Along such a turn, the absolute value of the deflection of $\mathcal{A}$ (its change of orientation) is greater than $\theta_{lim} = \kappa_{max}^2/\sigma_{max}$.

Figure 2 shows the curvature profile of a *lsr* SCC-path, i.e. a SCC-path made of a left turn, a line segment and a right turn. This curvature profile is a piecewise linear function of the arc length, each linear part corresponding to a different arc: the segment is the null part (#4), the circular arcs are the non-null constant parts (#2 and #6) and the clothoid arcs are the other linear parts (#1, #3, #5 and #7).

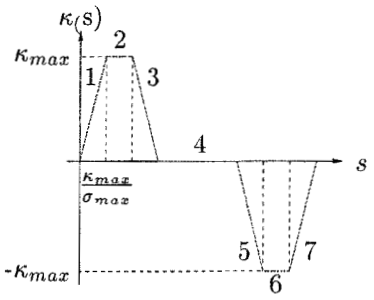

Figure 2: the curvature profile of a SCC-path.

We are searching the set of the final configuration of the continuous-curvature turns starting at a given configuration with null curvature. As this configuration has a null curvature, it can be made equal to the null configuration (by translation and rotation). In the following, we will consider a left turn but a similar reasoning can be done for a right turn.

We call $T_l(\beta)$ the left turn of deflection β starting at the null configuration, for $\beta \in [\theta_{lim}, 2\pi[$ (turns of deflection smaller than θ_{lim} do not exist). The first arc of $T_l(\beta)$ is a clothoid arc of sharpness σ_{max} and length $\kappa_{max}/\sigma_{max}$. Its final configuration $q_1(\beta)$ is:

$$q_1(\beta) = q_1 = \begin{pmatrix} x_1 &= \sqrt{\pi/\sigma_{max}} FrC\left(\sqrt{\frac{\theta_{lim}}{\pi}}\right) \\ y_1 &= \sqrt{\pi/\sigma_{max}} FrS\left(\sqrt{\frac{\theta_{lim}}{\pi}}\right) \\ \theta_1 &= \theta_{lim}/2 \\ \kappa_1 &= \kappa_{max} \end{pmatrix}$$

with FrC and FrS, the Fresnel integrals, defined respectively as:

$$FrC(x) = \int_0^x \cos\frac{\pi}{2}u^2 du, \quad FrS(x) = \int_0^x \sin\frac{\pi}{2}u^2 du,$$

The second piece of $T_l(\beta)$ is a circular arc of radius $1/\kappa_{max}$ starting at q_1, of length $(\beta - \theta_{lim})/\kappa_{max}$. This circular arc lies on the circle C_l of center $\Omega_l = (x_1 - \sin\theta_1/\kappa_{max}, y_1 + \cos\theta_1/\kappa_{max})$ (see fig. 3), and ends at the configuration $q_2(\beta)$. The last piece of $T_l(\beta)$ is the clothoid of sharpness - σ_{max} starting at $q_2(\beta)$ and of length $\kappa_{max}/\sigma_{max}$. This clothoid, and the turn, finishes at the configuration $q_3(\beta)$.

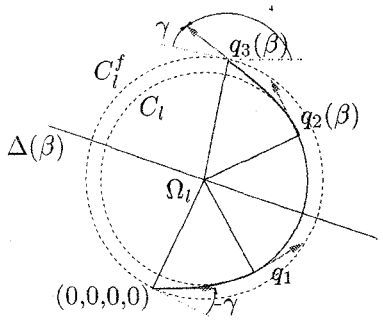

Figure 3: a continuous-curvature left turn.

As the curvature profile κ of this turn is a symmetric function ($\kappa(s) = \kappa(l-s), \forall s \in [0,l]$, where $l = (\beta + \theta_{lim})/\kappa_{max}$ is the length of $T_l(\beta)$), so is the curve of $T_l(\beta)$ in $\mathcal{W}$: this curve is symmetric w.r.t. the line $\Delta(\beta)$ of direction $(\beta+\pi)/2$ and containing Ω_l. Thus, the distance between the position of $q_3(\beta)$ and Ω_l remains equal to $R_T = \sqrt{x_{\Omega_l}^2 + y_{\Omega_l}^2}$, the distance between the position of the starting configuration of the turn and Ω_l: the position of $q_3(\beta)$ remains on the circle C_l^f of center Ω_l and radius R_T when β belongs to $[\theta_{lim}, 2\pi[$.

Moreover, this symmetry also implies that the angle γ between the tangent to this circle C_l^f at the position of q_3 and the line containing this position and of orientation θ_3 is the opposite of the angle between the tangent to the circle C_l^f at the null position and the x-axis (see fig. 3): $\gamma = -\arctan(x_{\Omega_l}/y_{\Omega_l})$. As a conclusion, the coordinates of the final configuration $q_3(\beta)$ of the left turn $T_l(\beta)$ are, for $\beta \in [\theta_{lim}, 2\pi[$:

$$q_3(\beta) = \begin{pmatrix} x_3(\beta) &=& R_T[\sin(\beta-\gamma) - \sin\gamma] \\ y_3(\beta) &=& R_T[\cos\gamma - \cos(\beta-\gamma)] \\ \theta_3(\beta) &=& \beta \\ \kappa_3 &=& 0 \end{pmatrix}$$

Using a left turn as defined before, the deflection of $\mathcal{A}$ (its change of orientation) is at least θ_{lim}. If $\mathcal{A}$ needs to turn of an angle β included in $]0, \theta_{lim}$, it will use an *elementary path* [17, 18] made of:

1. a first clothoid arc of sharpness $\sigma \in]0, \sigma_{max}]$ and of length $l \in [0, \kappa_{max}/\sigma]$, and
2. a second clothoid arc of sharpness -σ and of length l.

The elementary paths of sharpness σ_{max} cannot be considered: their final configuration are located on a parametric curve, whose equation includes the Fresnel integrals FrC and FrS. This curve is too complex to be used. Thus, we choose the sharpness of the elementary path so that the final configuration of this turn is located on C_l^f, the same circle as the final configuration of the left turns of deflection greater than θ_{lim}. To finish at the configuration $(R_T[\sin(\beta-\gamma) - \sin\gamma], R_T[\cos\gamma - \cos(\beta-\gamma)], \beta, 0)$, for $\beta \in]0, \theta_{lim}[$, the elementary path must have for sharpness (using the formulas of [17, 18]):

$$\sigma = \frac{\pi\left(\cos\frac{\beta}{2}\, FrC\left(\sqrt{\frac{\beta}{\pi}}\right) + \sin\frac{\beta}{2}\, FrS\left(\sqrt{\frac{\beta}{\pi}}\right)\right)^2}{R_T^2 \sin^2(\beta/2 - \gamma)}$$

and for length $l = \sqrt{\beta/\sigma}$.

We must verify if this elementary path exists and is feasible, i.e. if $0 < \sigma \leq \sigma_{max}$ and $\sigma l \leq \kappa_{max}$. As long as $\theta_{lim} = \kappa_{max}^2/\sigma_{max}$ is smaller than $\theta_{lim}^{max} \approx 1.4626\pi$ ($3\pi/4 < \theta_{lim}^{max} < 2\pi/3$), σ is defined for $\beta \in]0, \theta_{lim}]$ (as $\theta_{lim}/2 - \gamma < \pi$) and is in $]0, \sigma_{max}]$. The maximum curvature along this turn is $\sqrt{\sigma\beta}$, which is smaller than $\sqrt{\sigma_{max}\theta_{lim}} = \kappa_{max}$. In our experiments, σ_{max} and κ_{max} are fixed w.r.t. real values of car-like vehicles, and θ_{lim} is always less than $\pi/3 \ll \theta_{lim}^{max}$.

To conclude, we have defined for each $\beta \in]0, 2\pi[$ a left continuous-curvature turn of deflection β starting at the null configuration. The final configuration of these left continuous-curvature turns are located on a circle C_l^f depending only of σ_{max} and κ_{max}. The same work can be achieved for the right continuous-curvature turns (the resulting sharpness are opposite, the lengths are the same, the circle C_r^f is the symmetric of C_l^f w.r.t. the x-axis). Moreover, these turns can be translated and rotated in order to start from any given null-curvature configuration. At last, the set of turns finishing at a given configuration q can be deduced from the set of turns starting at the configuration opposed to q: they have the same curve in $\mathcal{W}$, with opposite direction.

4.3 Computing SCC-Paths

In this section, we will describe how the at most six SCC-paths linking q_a to q_b, two null-curvature configurations, are defined.

First, let us consider the paths made of three turns, e.g. the path *lrl* (the path *rlr* can be defined similarly).

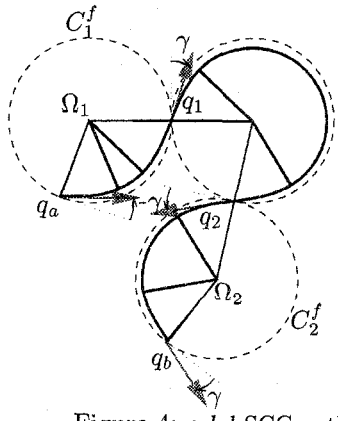

Figure 4: a *lrl* SCC-path.

1. Let C_1^f be the circle of the end configurations of the left turns starting from q_a, and Ω_1 be its center.

2. Let C_2^f be the circle of the start configurations of the left turns finishing at q_b, and Ω_2 be its center.
3. If $\Omega_1\Omega_2$ is smaller than $4R_T$, there exist at most two circles of radius R_T tangent simultaneously to C_1^f and C_2^f. For each C_i^f of these at most two circles, let q_1 and q_2 be the configurations of C_1^f and C_2^f respectively, which are located on C_i^f (see fig. 4). A *lrl* SCC-path linking q_a to q_b is made of the left turn linking q_a to q_1, the right turn linking q_1 to q_2 and the left turn linking q_2 to q_b. The shortest of the at most two possible paths is selected.

On a second hand, let us consider the paths made of a turn, a segment and a second turn, e.g. the path *lsr* (the three other paths, *lsl*, *rsl* and *rsr*, can be defined similarly).

1. Let C_1^f be the circle of the end configurations of the left turns starting from q_a, and Ω_1 be its center.
2. Let C_2^f be the circle of the start configurations of the right turns finishing at q_b, and Ω_2 be its center.

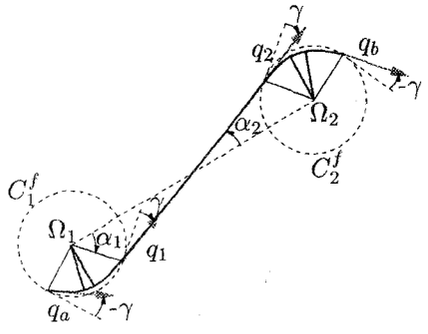

Figure 5: a *lsr* SCC-path.

3. We search the connecting line between C_1^f and C_2^f (see fig. 5); this line is the segment connecting two configurations of same orientation, q_1 of C_1 to q_2 of C_2, this orientation being equal to the orientation of the segment. Let α_1 be the angle between $\overrightarrow{\Omega_1\Omega_2}$ and $\overrightarrow{\Omega_1 q_1}$, and α_2 the angle between $\overrightarrow{\Omega_1\Omega_2}$ and the connecting line. For the turns *lsl* and *rsr*, α_2 is trivially 0, but for the turns *lsr* and *rsl*, the elementary properties of the triangle implies that $\sin\alpha_2 = 2\cos\gamma R_T/\Omega_1\Omega_2$. In both case, $\alpha_1 = \alpha_2 - \gamma - \pi/2$, which determines the connecting line.

   As α_1 must remain negative, in the case of the turns *lsr* and *rsl*, $\sin\alpha_2$ must remain smaller than $\cos\gamma$, which implies that the distance $\Omega_1\Omega_2$ between the two circles must remain greater than $2R_T$.

   If this inequality holds, the *lsr* path is then made of the left turn linking q_a to q_1, the segment q_1 q_2 and the right turn linking q_2 to q_b.

4.4 Collision Checking

The previous paths, once computed, must be checked for collision avoidance. The surface swept by $\mathcal{A}$ while following a segment or a circular arc can be explicitly calculated: it is a generalized polygon, i.e. a polygon whose sides are either linear segments or circular arcs. In this case, the collision detection with a given obstacle $\mathcal{B}$ is a simple intersection check between this surface and the polygonal region of $\mathcal{B}$. To have such a collision detection for clothoid arcs, we need to evaluate the surface swept by $\mathcal{A}$ along such an arc.

But a clothoid arc can only be represented by a parametric equation using Fresnel integrals, and these integrals can only be approximated. Therefore, the surface swept by $\mathcal{A}$ while following such an arc also has to be an approximation. This approximation is based on the concept of *motion polygon*. If there exists a clothoid arc linking a configuration q_a to a configuration q_b and turning left, the motion polygon from q_a to q_b, noted $\mathcal{MP}(q_a, q_b)$, is defined by the following points in clockwise order (see fig. 6):

- the rear right point of $\mathcal{A}(q_a)$;
- the rear left point of $\mathcal{A}(q_a)$;
- the front left point of $\mathcal{A}(q_b)$;
- the front right point of $\mathcal{A}(q_b)$;
- the intersection of the right side of $\mathcal{A}(q_a)$ and the right side of $\mathcal{A}(q_b)$;

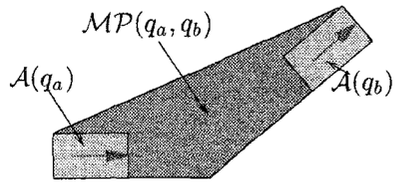

Figure 6: the motion polygon.

The motion polygon can be similarly defined for a clothoid arc turning right (i.e. if q_b is in the right half-plane defined by q_a). It can be proved (the proof is omitted due to lack of space) that this polygon has the following property: if $\mathcal{A}$ follows a clothoid arc from q_a to q_b, the surface $\cup_q \mathcal{A}(q)$ swept by $\mathcal{A}$ while in motion is included in $\mathcal{MP}(q_a, q_b)$.

Therefore, if a clothoid arc linking two configurations q_s to q_f is approximated by a set of configurations $\{q_i, i \in \{0, \ldots, k\}\}$, with $q_s = q_0$ and $q_f = q_k$, the surface swept by $\mathcal{A}$ along the clothoid arc is approximated by $\cup_{i\in\{1,\ldots,k\}} \mathcal{MP}(q_{i-1}, q_i)$. The more precise the discretization of the clothoid arcs is, the more precise the approximation of the surface swept along a clothoid arc is.

5 The Global Path Planner

In 1995, Svestka and Overmars presented a general planning scheme called the Probabilistic Path Planning (PPP) which allows to build a (probabilistically complete) global planner using a local planner [21].

For a given robot and workspace, the PPP method consists in two phases. During the first one, called the *learning phase*, it builds a roadmap and stores this map in a graph (the *exploration graph*) whose nodes are collision-free configurations and edges are (simple) feasible paths, or *local paths*, computed by the local planner. In the second phase, the *query phase*, given a start configuration q_s and a goal

one q_g, PPP tries to connect these configuration to the exploration graph using the local planner, and then performs a graph search. Therefore, when PPP finds a path linking q_s to q_g, the result path is a concatenation of local paths.

We are currently using this path planning scheme because the exploration graph can be reused for several new planning problem, as long as the robot, the workspace and the obstacles considered remain the same. Moreover, this graph may be extended in order to plan multi-vehicles' motion, or trajectories among moving obstacles (these possibilities are considered for future works). However, any other path planning scheme which uses a local planner independently of the way this planner works (as e.g. the Ariadne's Clew algorithm [13]) could be considered to obtain a global continuous-curvature path planner.

6 Experimental Results

The local and global path planners have been implemented in C++ using LEDA, a geometric and algorithmic library [2].

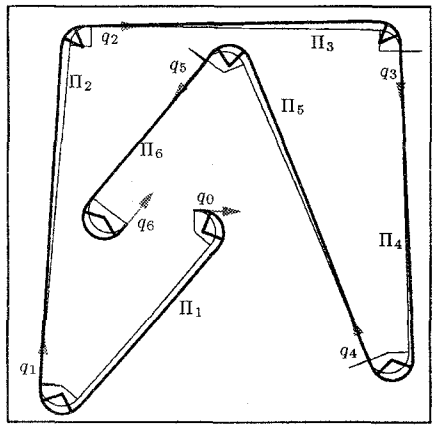

Figure 7: Dubins' curves vs. **SCC-paths.**

To begin with, the local path planner has been tested in an obstacle-free workspace. The purpose here was to compare the respective lengths of SCC-paths and Dubins' curves. Figure 7 depicts the SCC-paths (thick lines) and the Dubins' curves (thin lines) linking a sequence of configurations $q_1, q_2 \ldots q_6$, while the paths' lengths (in meter) are listed in Table 1. In this experiment, the curves have a maximum curvature of 0.2 m^{-1} (the minimum turning radius is 5 m), and the SCC-paths have a maximum sharpness of 0.05 m^{-2} (the minimum deflection to reach the maximum curvature is $\theta_{lim} = 0.8$ radian ≈ 45.84 degrees).

Path	Π_1	Π_2	Π_3	Π_4	Π_5	Π_6
Dubins	71.67	89.47	76.19	86.06	87.70	58.90
SCC	78.35	91.37	78.24	90.01	90.86	63.01

Table 1: lengths of Dubins' curves and SCC-paths.

The global path planner was then tested in two environments taken from Laumond et al. [12] respectively containing four and five obstacles. Both correspond to a 40 m sided square workspace with a 2.5 m long and 1.5 m wide car-like vehicle. Unlike Laumond et al., **forward only** paths were generated. Various experimental results are depicted in Figs. 8 and 9.

[2] LEDA has been developed by the Max-Planck-Institut fuer Informatik Im Stadtwald (Saarbruecken, DE)

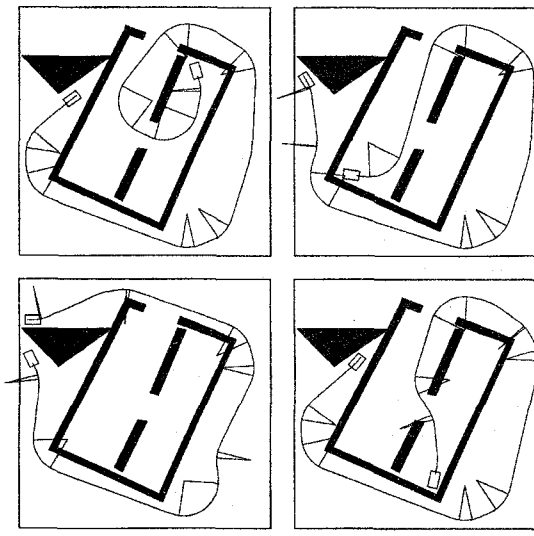

Figure 8: experiments with 4 obstacles.

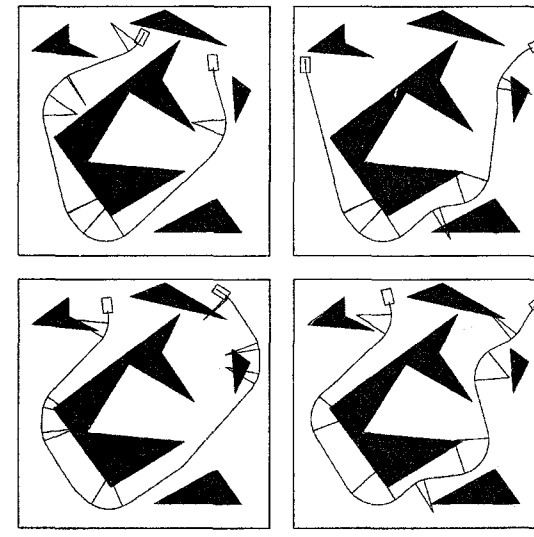

Figure 9: experiments with 5 obstacles.

7 Conclusion and Discussion

In this paper, we have considered path planning for a car-like vehicle. Previous solutions to this problem would compute paths made up of circular arcs connected by tangential line segments. Such paths have a non continuous curvature profile. Accordingly a vehicle following such a path has to stop at each curvature discontinuity in order to reorient its front wheels. To remove this limitation, we added a continuous-curvature constraint to the problem at hand. In addition, we introduced a constraint on the curvature derivative so as to reflect the fact that a car-like vehicle can only reorient its front wheels with a finite velocity.

We proposed a solution to the problem at hand that relies upon the definition of a set of paths with continu-

ous curvature and maximum curvature derivative. These paths contain at most eight pieces, each piece being either a line segment, a circular arc of maximum curvature, or a clothoid arc. They are called *SCC-paths* (for Simple Continuous Curvature paths). They have been used to design a *local path planner*, i.e. a non-complete collision-free path planner, which in turn has been embedded in a global path planning scheme. The result is the first path planner for a car-like vehicle that generates collision-free paths with continuous curvature and maximum curvature derivative. Experimental results have been presented. Future works will explore three main directions:

1. Demonstrate the sub-optimal nature of SCC-paths. Optimal paths for the problem at hand are likely to be made of an infinity of pieces (see [10, 11]. We would do so by explicitly computing an upper bound of the difference between the length of the SCC-path and the Dubins' curve linking two given configurations, and proving that this bound tends towards zero when $\sigma_{max} \rightarrow \infty$ for a given κ_{max}.
2. Extend the path planner to the case of a car-like vehicle that can move both forward and backward. This is straightforward; Reeds and Shepp's curves [16] would be used rather than Dubinses.
3. Compute the partition of the configuration space with respect to the type of the shortest SCC-path (like Bui et al. [3] did for Dubins' curves). In fact, the six possible paths linking two configurations should be ordered (with increasing length) a priori, before their computation. Thus, the best collision-free path linking two configurations could be found with a minimal cost.

Acknowledgments

This work was partially supported by the Inria-Inrets[3] Praxitèle programme on urban public transport *[1994-1997]*, and the Inco-Copernicus ERBIC15CT960702 project "Multi-agent robot systems for industrial applications in the transport domain" *[1997-1999]*.

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. Cerezo, 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, January 1994.

[3] X.-N. Bui, P. Souères, J.-D. Boissonnat, and J.-P. Laumond. Shortest path synthesis for Dubins non-holonomic robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 2–7, San Diego (CA), May 1994.

[4] H. Delingette, M. Hébert, and K. Ikeuchi. Trajectory generation with curvature constraint based on energy minimization. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 206–211, Osaka (JP), November 1991.

[5] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[3] Inst. Nat. de Recherche sur les Transports et leur Sécurité.

[6] S. Fleury, Ph. Souères, J.-P. Laumond, and R. Chatila. Primitives for smoothing paths of mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 832–839, Atlanta (GA), September 1993.

[7] Y. Kanayama and B. I. Hartman. Smooth local path planning for autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1265–1270, Scottsdale (AZ), May 1989.

[8] Y. Kanayama and N. Miyake. Trajectory generation for mobile robots. In *Proc. of the Int. Symp. on Robotics Research*, pages 16–23, Gouvieux (FR), 1985.

[9] K. Komoriya and K. Tanie. Trajectory design and control of a wheel-type mobile robot using B-spline curve. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 398–405, Tsukuba (JP), September 1989.

[10] V. Kostov and E. Degtiariova-Kostova. Some properties of clothoids. Research Report 2752, Inst. Nat. de Recherche en Informatique et en Automatique, December 1995.

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

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

[13] E. Mazer, J.-M. Ahuactzin, E.-G. Talbi, and P. Bessière. Robot motion planning with the Ariadne's Clew algorithm. In *Proc. of the Int. Conf. on Intelligent Autonomous Systems*, Pittsburgh (PA), February 1993.

[14] B. Mirtich. Using skeletons for nonholonomic motion planning. Research Report ESRC 92-16/RAMP 92-6, Departement of Electrical Engineering and Computer Science, University of California, Berkeley, June 1992.

[15] W. L. Nelson. Continuous curvature paths for autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1260–1264, Scottsdale (AZ), May 1989.

[16] 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.

[17] A. Scheuer and Th. Fraichard. Planning continuous-curvature paths for car-like robots. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1304–1311, Osaka (JP), November 1996.

[18] A. Scheuer and Th. Fraichard. Collision-free and continuous-curvature path planning for car-like robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 867–873, Albuquerque (NM), April 1997.

[19] A. Takahashi, T. Hongo, and Y. Ninomiya. Local path planning and control for AGV in positionning. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 392–395, Tsukuba (JP), September 1989.

[20] P. Švestka and M. H. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1631–1636, Nagoya (JP), May 1995.

[21] P. Švestka and M. H. Overmars. Probabilistic path planning. Technical Report UU-CS-1995-22, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands, May 1995.