Smooth trajectory planning for a car-like vehicule in a structured world

Thierry Fraichard, C. Laugier

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1448

Programme 4
Robotique, Image et Vision

SMOOTH TRAJECTORY PLANNING FOR A CAR-LIKE VEHICLE IN A STRUCTURED WORLD

Thierry FRAICHARD
Christian LAUGIER

GROUPE DE RECHERCHE
GRENOBLE

Juin 1991

RR-1448

Smooth trajectory planning for a car-like vehicle in a structured world*

Planification de trajectoire sans manœuvre pour un véhicule de type voiture en environnement structuré

Thierry Fraichard Christian Laugier†
IMAG–LIFIA Laboratory
46, av. Félix Viallet, 38031 Grenoble Cedex, France
E-mail: tf@lifia.imag.fr, laugier@lifia.imag.fr

*The work presented here is part of the SHARP project within the framework of IRIMAG (INRIA, CNRS, INPG, UJF).
†Research director at INRIA.

Résumé

Ce rapport étudie la planification de trajectoire pour un véhicule de type voiture—c.a.d. un véhicule non holonome à rayon de giration borné—dans un environnement statique et structuré. En ce qui concerne la structure de l'environnement, nous supposons l'existence de voies de circulation naturelles à l'intérieur desquelles le véhicule peut se déplacer. La contribution de ce rapport est un algorithme de planification de trajectoire qui, à partir de la ligne polygonale $\mathcal{S}$ représentant l'axe de la voie de circulation que le véhicule doit suivre, engendre une trajectoire $\mathcal{C}$ qui évite les obstacles de l'environnement, qui est sans manœuvre et qui est exécutable par le véhicule compte tenu de ses contraintes cinématiques. En outre, $\mathcal{C}$ est "topologiquement équivalente" à $\mathcal{S}$; autrement dit, $\mathcal{C}$ demeure dans la voie de circulation définie par $\mathcal{S}$. $\mathcal{C}$ est composée de segments de droite et d'arcs de cercle tangents d'un rayon r donné. Le principe de base de l'algorithme consiste à rechercher le centre de chaque arc de cercle dans un domaine particulier appelé "espace des centres de courbure" [9]. Cet algorithme est efficace; sa complexité est $\mathcal{O}(nm)$ où n représente le nombre d'obstacles dans l'environnement et m le nombre de segments dans $\mathcal{S}$. Cette efficacité est obtenue aux dépens de la complétude. En effet, l'algorithme utilise une heuristique qui n'est pas complète. Cependant, de nombreux tests en environnement de type réseau routier se sont révélés concluants.

Mots clés: robots mobiles, planification de trajectoire, non holonomie, trajectoire sans manœuvre.

Abstract

This report aims at studying the trajectory planning for a car-like vehicle —i.e. a non-holonomic vehicle whose turning radius is lower bounded—in a static and structured world. As for the structure of the world, we assume the existence of natural lanes within which the vehicle is able to move. The contribution of this report is a smooth trajectory planning algorithm which, when given the polygonal line $\mathcal{S}$ representing the spine of the lane that the vehicle has to follow, generates a trajectory $\mathcal{C}$ which avoids the obstacles of the world and which is smooth—i.e. without backing up manœuvres—and executable by the vehicle according to its own kinematic constraints. Besides $\mathcal{C}$ is 'topologically equivalent' to $\mathcal{S}$; in other words, $\mathcal{C}$ must remain in the lane defined by $\mathcal{S}$. $\mathcal{C}$ is made up of straight segments and tangential circular arcs of a given radius r. The basic principle of the algorithm is to search for the centre of each circular arc in a particular domain called 'curvature centres space' [9]. This algorithm is efficient; its computational complexity is $\mathcal{O}(nm)$ where n is the number of obstacles in the world and m the number of segments in $\mathcal{S}$. This efficiency is obtained at the expense of completeness because the algorithm makes use of a heuristic which is not complete. However tests in roadway-like environments have proved successful.

Key words: mobile robots, path planning, non-holonomy, smooth path.

Contents

1 Introduction

1.1 General presentation

This report aims at studying the trajectory planning for a car-like vehicle—i.e. a non-holonomic vehicle whose turning radius is lower bounded—in a static and structured world. As for the structure of the world, we assume the existence of natural lanes within which the vehicle is able to move. These lanes are defined by the intrinsic structure of the world. Such a lane is represented by its spine. Given the spine $\mathcal{S}$ of a lane, we are interested in planning a trajectory $\mathcal{C}$ which avoids the obstacles of the world and which is smooth—i.e. without backing up manœuvres—and executable by the vehicle according to its own kinematic constraints. Besides $\mathcal{C}$ must be 'topologically equivalent' to $\mathcal{S}$; in other words, $\mathcal{C}$ must remain in the lane defined by $\mathcal{S}$. The framework of this study is the European Prometheus Eureka project whose purpose is to design new cars and new road infrastructures. The road network is obviously a highly structured environment. It is designed to ease the traffic; for instance, when a vehicle intend to cross an intersection, it is supposed to follow a particular lane. Our contribution to this project is to build software tools to guide several car-like vehicles moving in subsets of the road network [3, 4].

1.2 Related works

Trajectory planning is certainly one of the most studied problems in robotics. The well-known 'piano mover' paradigm, in which the trajectory planning problem for any mobile is turned into the trajectory planning of a point in some space called the configuration space[1], brought about lots of papers proposing general or specific, exact or approximate, and efficient or inefficient, methods to solve this problem (see [15] for a recent overview). With this formulation, the existence of a collision-free trajectory for the mobile is characterized by the existence of a connected component in the admissible—i.e. collision-free—configuration space. However many classical methods prove inadequate when the mobile is subject to a non holonomic kinematic constraint—i.e. a constraint expressed as a non-integrable equation involving the derivatives of the configuration parametres. Since this equation is not integrable, there are constraints in the tangent space at each configuration (i.e. on the allowable velocities). The main consequence of a non-holonomic constraint is that an

[1]The *configuration* of a body A is a set of independent parametres that characterize the position and orientation of every point in A. Let k be the number of parametres required to specify the configuration of A. The configuration of A can be regarded as a point in a k-dimensional space. This k-dimensional space is the *configuration space* of A.

arbitrary path in the admissible configuration space does not necessarily correspond to a feasible trajectory for the mobile.

Dealing with non holonomy in trajectory planning is relatively recent. But it has already produced some important theoretical and practical results. [8] studies the problem for a car-like mobile and give the condition of existence of a trajectory for such a mobile. This first result is implemented in [12]. Other contributions are also presented in [1, 14] and [17]. Important results have been obtained by attacking the problem with tools from non-linear control and differential geometry (see [11] for a presentation of these tools). The most important result is the 'controllability' theorem; it states that for a non-holonomic mobile which is 'controllable', the existence of a collision-free trajectory is characterized by the existence of a connected component in the free—i.e. collision-free and contact-free—configuration space. As a consequence of the controllability theorem, a collision-free trajectory for a non-holonomic controllable mobile can be derived from a solution for the associated holonomic mobile. This idea is the basis of the planners presented in [6], [13] and [7].

More interesting for our problem are the approaches aiming at planning smooth—i.e. manœuvre-free—trajectories. This problem appears to be more difficult than the case in which manœuvres are allowed. Indeed there is no such controllability result. [2] presents a decision algorithm which decides if a smooth path exists but this algorithm is not constructive. Various planning algorithms are proposed in [5, 9] and [18]. All these methods generate trajectories made up of straight segments connected with tangential circular arcs. The methods described in [5] and in [9] deal with a circular mobile while the approach presented in [18] consider a mobile which is, this time, rectangular but compelled to follow some predefined straight lines.

1.3 Contribution of the report

The contribution of this report is a smooth trajectory planning algorithm for a rectangular car-like vehicle which, when given the polygonal line $\mathcal{S}$ representing the spine of the lane that the vehicle has to follow, generates a trajectory $\mathcal{C}$ made up of straight segments and tangential circular arcs which meets the constraints mentioned earlier. Let us notice that unlike [18], the vehicle is not compelled to follow the spine of the lane. The basic principle of the algorithm is to search for the centre of each circular arc in a particular domain called 'curvature centres space'. Computing and searching this bidimensional domain is carried out by using a method derived from the one described in [9] and adapted according to the characteristics of our problem—i.e. dealing with a rectangular vehicle and satisfying the 'topological equivalence' property. This algorithm is efficient, its computational complexity is $\mathcal{O}(nm)$ where n is the number of obstacles in the world and m the number of segments in the spine $\mathcal{S}$. This efficiency is obtained at the expense of completeness because the algorithm makes use of a heuristic which is not complete—i.e. it may fail to find a solution even if there is one. However tests in roadway-like environments have proved successful.

The report is organized as follows: §2 presents the description of the environment and the modelling of the vehicle. §3 briefly outlines the algorithm which is completely presented in §4. Finally experimental results are presented in §5.

2 The world model

2.1 The environment

The natural workspace $\mathcal{W}$ of a car-like vehicle is the roadway. Since this roadway is made up of smooth bidimensional surfaces, it is possible to map these surfaces on a plane when reasoning on the vehicle's motions. Therefore, we consider $\mathcal{W}$ to be the plane R^2. $\mathcal{W}$ is cluttered with a set $\mathcal{O}$ of static obstacles representing the limits of the roadway (verges, pavements, central reservations). Each obstacle $o \in \mathcal{O}$ is represented by a generalized polygon i.e. by a polygon whose edges are either straight segments or circular arcs [10].

2.2 The vehicle

2.2.1 Kinematic characteristics

Let M be a car-like vehicle. Such a vehicle has two rear wheels and two directional front wheels. M is modelled by a bidimensional rigid rectangle translating and rotating in the plane. A configuration of M is defined by the t-uple (x_r, y_r, θ) of $R^2 \times S^1$—S^1 is the oriented unit circle—where x_r and y_r are the coordinates of the rear axle midpoint R and θ is the orientation of M—i.e. the angle between the x axis of the cartesian frame embedded in the plane and the main axis of the vehicle (figure 1).

Let us consider that M is front wheel driven and that the associated velocity vector is applied at the front axle midpoint F. Then the control parametres of the car are the module v of the velocity vector of F and the steering angle ϕ measuring the orientation of the velocity vector of F with respect to the main axis of M. The motion equations relating the control parametres (v, ϕ) and the configuration parametres (x_r, y_r, θ) of M are:

$$\left.\begin{array}{rcl}\dot{x}_r &=& v\cos\theta\cos\phi \\ \dot{y}_r &=& v\sin\theta\cos\phi \\ \dot{\theta} &=& v\sin\phi/l_w\end{array}\right\}$$

where l_w is the wheelbase of the vehicle i.e. the distance between R and F. From this set of equations, we can easily deduce the following relation:

$$\theta = \arctan(\dot{y}_r/\dot{x}_r) \tag{1}$$

This non-integrable constraint involving the derivatives of the configuration parametres is non-holonomic. It states that the vehicle can only move in a direction tangent to its orientation. Therefore it prevents M from executing some particular trajectories (a pure rotation for instance).

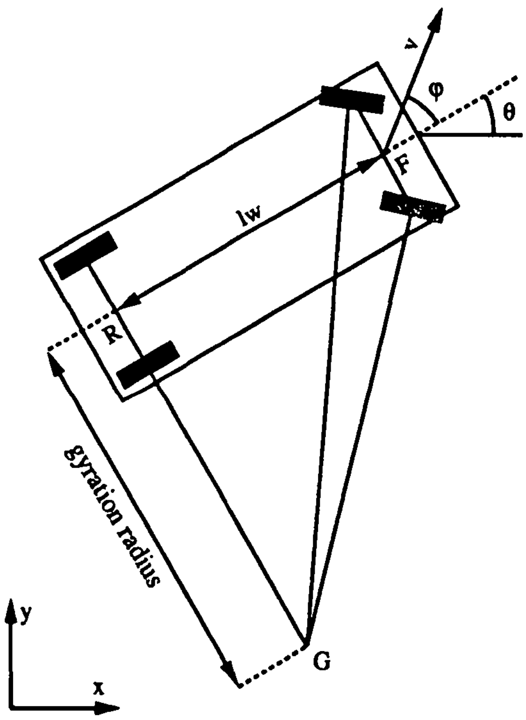

Figure 1: a car-like vehicle

Besides M is designed so that, during a turn, the axles of the front wheels will intersect the rear axle at a particular point G. This point is the instantaneous gyration centre of M and the distance from R to G is the instantaneous curvature radius of the executed trajectory (figure 1). This curvature radius is lower bounded by a certain value r depending both on the maximum steering angle and the maximum centrifugal acceleration tolerated by the vehicle when in motion (this acceleration must usually be limited to the order of magnitude $1g$). Let v_{max} and ϕ_{max} be respectively the maximum velocity of the vehicle and the maximum steering angle, $r = \min(v_{max}^2, l_w/\tan\phi_{max})$.

2.2.2 Trajectory characteristics

The trajectory planning of a rigid body in Euclidean space is classically turned into the trajectory planning of a point in some space called the configuration space[1]. In our context, planning a trajectory for M in $\mathcal{W} = R^2$ is equivalent to planning a trajectory for its reference point R in its associated three-dimensional configuration space $CS = R^2 \times S^1$. Such a trajectory computed in CS must meet the kinematic constraints mentioned in §2.2.1 in order to be executable by M in $\mathcal{W}$.

However the relation (1) shows that there is a one-to-one correspondence between a trajectory in CS and its projection in $\mathcal{W}$. Therefore any bidimensional curve $\mathcal{C}$ of $\mathcal{W}$ defines a particular trajectory for M completely. In order to represent a trajectory which is executable by M, $\mathcal{C}$ must meet the two following properties:

1. $\mathcal{C}$ is piecewise of class C^2—a curve is of class C^n if it is differentiable n times and if its n^{th} derivative is continuous—(non-holonomy constraint).

2. The curvature in each point of $\mathcal{C}$ is less than $1/r$ where r is the minimum gyration radius.

Besides since we are interested in finding smooth trajectories i.e. without backing up manœuvres, $\mathcal{C}$ must also meet the following property:

3. $\mathcal{C}$ is of class C^1.

3 Outline of the trajectory planner

3.1 General presentation

The inputs of the trajectory planner are:

1. The geometric description of the static obstacles of $\mathcal{W}$ i.e. the set $\mathcal{O}$ of generalized polygons.

2. The geometric description of the vehicle M.

3. The minimum curvature radius r of M. As mentioned in §2.2.1, r is a function both of the vehicle maximum steering angle and of the maximum speed allowed.

4. The spine $\mathcal{S}$ of the lane that M has to follow. $\mathcal{S}$ is a polygonal line represented by an ordered set of points $(p_1, p_2 \dots p_n)$. $\mathcal{S}$ is assumed to meet the following properties:

   (a) $\forall\ o \in \mathcal{O}, \forall\ i = 1 \dots n-1$, the segment $p_i p_{i+1}$ does not intersect $\mathcal{G}(o, w/2)$ where $\mathcal{G}(o, w/2)$ represents the obstacle o isotropically grown of the half width $w/2$ of M. In other words, M can follow any straight segment p_i, p_{i+1} of $\mathcal{S}$ with the orientation $\overrightarrow{p_i p_{i+1}}$ without generating any collision (except maybe at the ends of the segment).

   (b) M is able to make a right-hand (resp. left-hand) turn starting from p_1 without generating a collision with an obstacle located on the left (resp. right) side of the segment $p_1 p_2$.

   (c) M is able to reach p_n through a right-hand (resp. left-hand) turn without generating a collision with an obstacle located on the left (resp. right) side of the segment $p_{n-1} p_n$.

   The hypothesis (4a) is sensible since p_i, p_{i+1} represents the axis of a lane within which M is supposed to be able to move. The purpose of the hypotheses (4b) and (4c) will be cleared up in §4.4.

As mentioned earlier, the purpose of the trajectory planner is to produce a trajectory which is collision-free and 'topologically equivalent' to $\mathcal{S}$ and executable by M. In our context, $\mathcal{C}$ is said to be topologically equivalent to $\mathcal{S}$ if and only if it has the same end points p_1 and p_n. the same tangent direction at p_1 and p_n and if it is homotopic[2] to $\mathcal{S}$ in $\mathcal{W}$. M starts from the position p_1 with the orientation $\overrightarrow{p_1 p_2}$ and it is to reach the position p_n with the orientation $\overrightarrow{p_{n-1} p_n}$.

The output of the trajectory planner is a geometric trajectory $\mathcal{C}$ made up of straight segments and of circular arcs of radius r tangentially connected so that the resulting curve will be of class C^1. Thus all the kinematics constraints of M presented in §2.2.2 are met. Let us notice that $\mathcal{C}$ is derived from $\mathcal{S}$ but does not necessarily include a subpart of each segment $p_i p_{i+1}$.

[2]Let f and g be the characteristic representations of two trajectories i.e. two continuous mappings from $[0, 1]$ into $\mathcal{W}$. f and g are *homotopic* if and only if there exists a continuous mapping Φ of $[0, 1] \times [a, b]$ into $\mathcal{W}$ such that $\Phi(t, a) = f(t)$ and $\Phi(t, b) = g(t)$.

3.2 Sketch of the algorithm

$\forall\, i = 2 \ldots n-1$, each t-uple (p_{i-1}, p_i, p_{i+1}) of the path $\mathcal{S}$ represents a transition from one straight segment to the next. Such a transition is called a 'turn' and is executed through a mere circular arc allowing M to move smoothly from the segment $p_{i-1}p_i$ to the segment p_ip_{i+1}. Thus the basic principle of the algorithm is to find for each turn of $\mathcal{S}$ the arc of radius r enabling the vehicle to execute the turn without any collision with the elements of $\mathcal{O}$. The algorithm determines the centre of this turning arc by building and searching a particular domain called 'curvature centres space'. This domain is built so that if the centre of the turning arc is picked up in it then the resulting trajectory for the turn will be collision-free and topologically equivalent to (p_{i-1}, p_i, p_{i+1}).

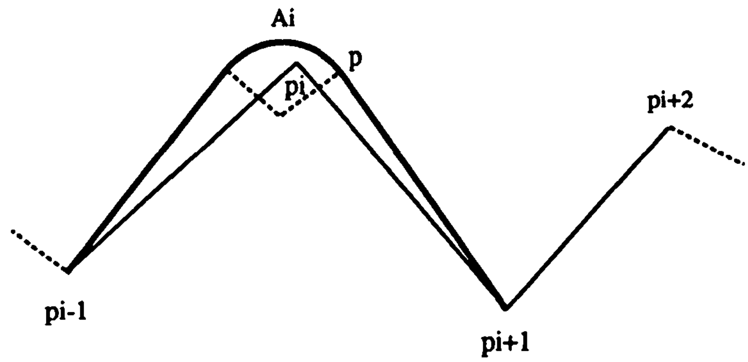

Figure 2: linking two consecutive turns

The full trajectory is determined by considering sequentially all the turns of $\mathcal{S}$ from the first to the last and by 'linking' together the consecutive turns. Figure 2 illustrates the linking mechanism. Let (p_{i-1}, p_i, p_{i+1}) be the t-uple associated with the i^{th} turn and let A_i be its associated turning arc. Linking together the i^{th} turn with the $i+1^{th}$ turn is carried out by considering the t-uple (p, p_{i+1}, p_{i+2})—where p is the point of A_i tangential to pp_{i+1}—as being the t-uple associated with the $i+1^{th}$ turn instead of (p_i, p_{i+1}, p_{i+2}).

4 The trajectory planner

4.1 Characterizing a turn

Let Λ be the current turn of $\mathcal{S}$ that the vehicle M has to execute. Λ is characterized by the t-uple $(\widehat{n_0n_1}, n_1n_2, n_2n_3)$ where $\widehat{n_0n_1}$ is the turning arc of centre c_p and of radius r_p associated with the previous turn; n_1n_2 and n_2n_3 are two connected segments (figure 3). Λ is a right-hand turn if $\overrightarrow{n_1n_2} \wedge \overrightarrow{n_2n_3} < 0$ and a left-hand turn otherwise. Remember that Λ has been previously computed so that $\forall\ o \in \mathcal{O}$, n_1n_2 and n_2n_3 will not intersect $\mathcal{G}(o, w/2)$.

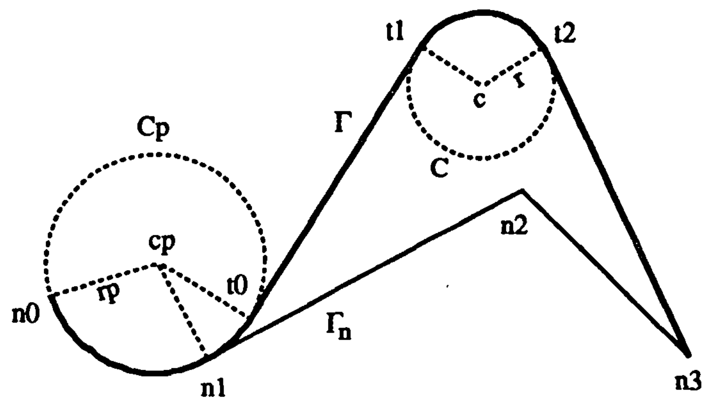

Figure 3: the turn Λ

Let Γ_n be the nominal trajectory associated with Λ, Γ_n is made up of the arc $\widehat{n_0n_1}$ and of the two segments n_1n_2 and n_2n_3. It represents a theoretical trajectory which is obviously not executable by M. The trajectory planner has to find a trajectory Γ which meets both the topological equivalence and the constraints expressed in §2.2.2. Thus the general form of Γ is a connected sequence $(\widehat{n_0t_0}, t_0t_1, \widehat{t_1t_2}, t_2n_3)$ of class C^1 (the arcs are connected tangentially to their associated segments); $\widehat{n_0t_0}$ is an arc of centre c_p and of radius r_p, $\widehat{t_1t_2}$ is an arc of radius r turning to the right or to the left according to Λ and t_0t_1 and t_2n_3 are two segments (figure 3).

4.2 A preliminary remark

A collision occurs if an obstacle intersects the region swept out by the vehicle M moving along a given trajectory $\mathcal{C}$. In other words, $\mathcal{C}$ does not collide with an obstacle $o \in \mathcal{O}$ if and only if the distance between $\mathcal{C}$ and o is greater than a certain value d depending on the geometric characteristics of both M and $\mathcal{C}$. Let w and l_f be respectively the width of M and the distance between R and the frontmost point of the vehicle. If $\mathcal{C}$ is a straight line then $d = w/2$. If $\mathcal{C}$ is a circular arc of radius r then $d = w/2$ for an obstacle o located inside the circle supporting the arc and $d = \mathcal{F}(r, l_f, w) = (\sqrt{(r + w/2)^2 + l_f{}^2} - r)$ for an obstacle o located outside the circle supporting the arc (figure 4). Therefore a trajectory $\mathcal{C}$ is collision-free if and only if it does not intersect $\mathcal{G}(o, d)$, $\forall\ o \in \mathcal{O}$. Since d is lower bounded by $w/2$, a necessary condition for $\mathcal{C}$ to meet the non-collision property is to deal with the obstacles of $\mathcal{O}$ isotropically grown of $w/2$. Let $\mathcal{O}'$ be the set of such obstacles.

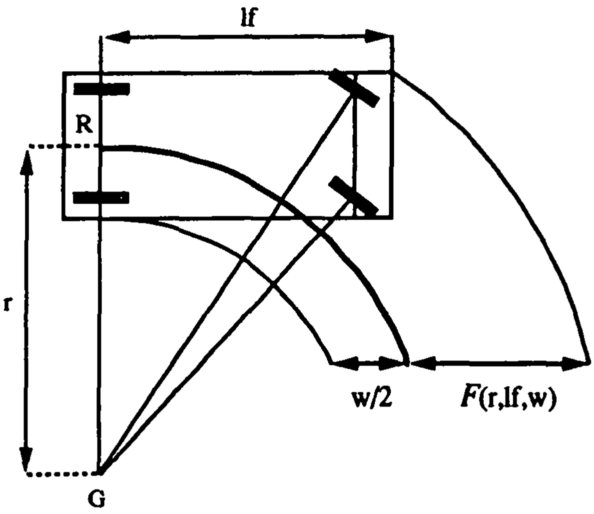

Figure 4: the region swept out by a turning vehicle

Note: this notion of 'swept out' region may be applied to any kind of vehicles—i.e. vehicles whose shape is not necessarily a rectangle. The algorithm presented in this report considers a rectangular vehicle but it could easily be extended to any polygonal vehicle.

Let us apply this non-collision criterion to the trajectory Γ. Let us define an 'external' obstacle as being an obstacle $o \in \mathcal{O}$ such that o is included in the concave domain bounded by $(t_0t_1, \widehat{t_1t_2}, t_2n_3)$. An obstacle which is not external is said to be 'internal'. Then a sufficient condition for the trajectory Γ to be collision-free with the obstacles of $\mathcal{O}$ is that Γ should not intersect the external obstacles grown of $\mathcal{F}(r, l_f, w)$ and the internal obstacles grown of $w/2$ (figure 5).

4.3 Building the curvature centres space for a general turn of $\mathcal{S}$

4.3.1 Defining D, D^* and $K(D^*)$

A turn is said to be general if it is neither the first nor the last turn of $\mathcal{S}$—i.e. it is assumed that there exists a previous turn and a next one as well.

Remember that Γ_n is the nominal trajectory associated with the current turn Λ. Let Γ_s be the shortest (according to the Euclidean distance) trajectory topologically equivalent to Γ_n and avoiding the obstacles of $\mathcal{O}'$. Γ_s is a geodesic curve i.e. an alternated sequence of contact free segments and contact arcs belonging to the boundaries of the obstacles of $\mathcal{O}'$ [16]. As the trajectory Γ

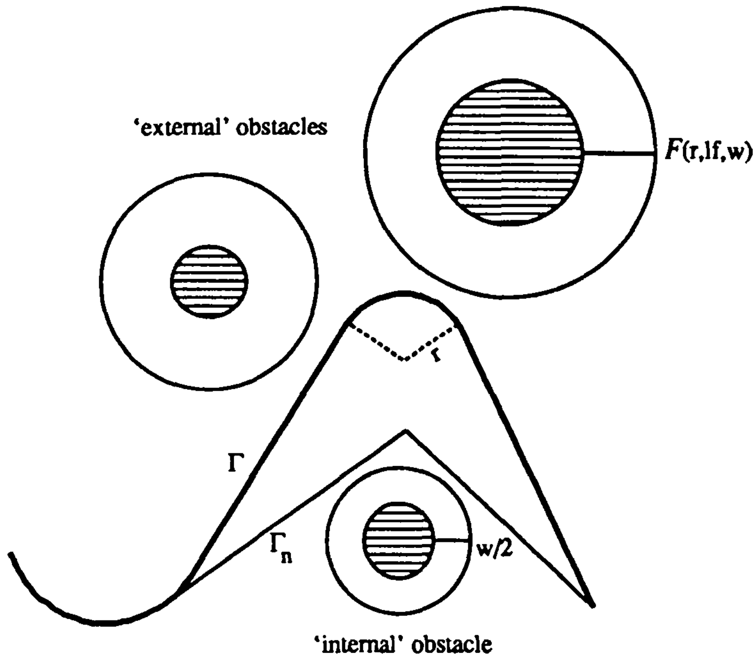

Figure 5: the non collision criterion

searched for must start with an arc $\widehat{n_0 t_0}$ of centre c_p and of radius r_p, the disk C_p supporting this arc is considered as a virtual obstacle at this step. Therefore Γ_s is made up of an arc of C_p starting from n_0, a segment S_1, a convex sequence A of segments and contact arcs and a segment S_2 ending at n_3 (figure 6a).

Let γ_s be the subpart of Γ_s made up of S_1, A and S_2. Let γ be the corresponding subpart of Γ. $\Gamma = \widehat{n_0 t_0} \cup \gamma$ where $\gamma = (t_0 t_1, \widehat{t_1 t_2}, t_2 n_3)$. According to our initial hypothesis of topological equivalence, γ must be homotopic to γ_s. Laumond shows in [9] that a trajectory homotopic to (S_1, A, S_2) lies in the concave domain D bounded by A and the half lines supporting S_1 and S_2. More precisely, if P_λ is the half plane bounded by the tangent to A whose direction is λ and which do not contain the contact obstacle then $D = \bigcup_{\lambda \in [\lambda_1, \lambda_2]} P_\lambda$ where λ_1 (resp. λ_2) is the orientation of S_1 (resp. S_2). Therefore the subpart γ of the trajectory Γ searched for must be located in the domain D (figure 6a). Let us notice that if Γ lies in the domain D then it is collision-free with respect to the obstacles located in the convex domain bounded by S_1, A and S_2.

Let γ^* be the dual line associated with γ—i.e. the polygonal line linking the arc centres of same curvature of γ. Laumond also shows in [9] that γ is homotopic to γ_s if and only if:

1. γ^* is contained in the domain $D^* = \bigcup_{\lambda \in [\lambda_1, \lambda_2]} P^*_\lambda$ where P^*_λ represents P_λ

translated of the vector $(r.\cos\lambda, -r.\sin\lambda)$ (figure 6b).

2. γ^* intersects the domain $K(D^*) = D^* \cap (\bigcap_{\lambda\in[\lambda_1,\lambda_2]} P^*_\lambda)$ (figure 6c).

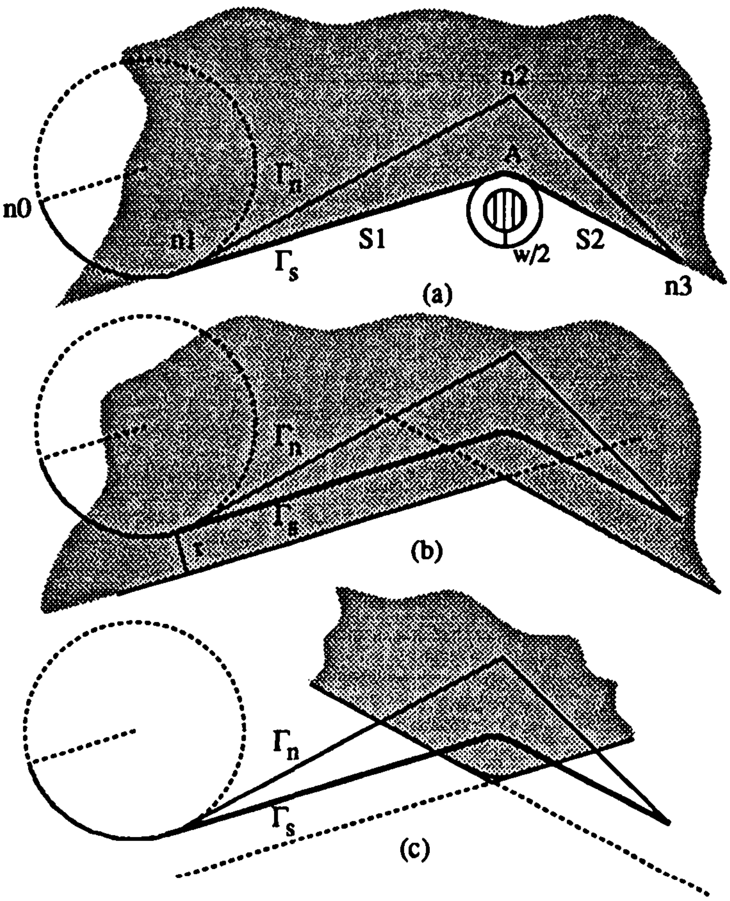

Figure 6: D, D^* and $K(D^*)$ for a turn

Since γ^* is here reduced to the centre c of $\widehat{t_1t_2}$, c must belong to $K(D^*)$. It stems from this property that the domain $K(D^*)$ can be used to characterize the set of the trajectories which are homotopic to γ_s. If a centre c is picked up in the domain $K(D^*)$ then the resulting trajectory Γ is topologically equivalent to Γ_s and therefore to Γ_n. Besides Γ is collision-free with respect to the obstacles located in the convex domain bounded by S_1, A and S_2.

4.3.2 Processing the obstacles included in D

The obstacles of $\mathcal{O}$ which are partly or completely located in D are still to be dealt with. Let $o \in \mathcal{O}$ be such an obstacle. It stems from the topological equivalence property holding between Γ_s and Γ that the area located between the concave domain defined by Γ_s and the convex domain defined by Γ is free of obstacles. Assuming that o is located in the concave domain defined by γ, there will be no collision between γ and o if the three following conditions are met:

1. The arc $\widehat{t_1t_2}$ must not intersect the obstacle $f(o) = \mathcal{G}(o, \mathcal{F}(r, l_f, w))$ (cf §4.2).

   This constraint can be expressed by removing from $K(D^*)$ the domain $\mathcal{G}(f(o), r)$. Indeed an arc of radius r does not intersect an obstacle if the centre of this arc is located at a distance greater than r of this obstacle.

2. The segment t_0t_1 must not intersect $\mathcal{G}(o, w/2)$.

   As a consequence, the point t_1 must not intersect the domain $g_a(o)$ which can be seen as the shadow of $\mathcal{G}(o, w/2)$ when lighted from t_0 (figure 7). This constraint can be expressed by removing from $K(D^*)$ the domain $\mathcal{G}(g_a(o), r)$.

3. The segment t_2n_3 must not intersect $\mathcal{G}(o, w/2)$.

   As a consequence, the point t_2 must not intersect the domain $g_b(o)$ which can be seen as the shadow of $\mathcal{G}(o, w/2)$ when lighted from n_3 (figure 7). This constraint can be expressed by removing from $K(D^*)$ the domain $\mathcal{G}(g_b(o), r)$.

As mentioned earlier, o has to be located in the concave domain bounded by γ. This constraint is met if γ lies outside the domain $g(o)$ which is defined as the convex hull of $g_a(o)$ and $g_b(o)$—i.e. the convex domain bounded by the half line E_1 supported by the line tangential to $\mathcal{G}(o, w/2)$ and passing through n_3, a part of the boundary of $\mathcal{G}(o, w/2)$ and the half line E_2 supported by the line tangential both to $\mathcal{G}(o, w/2)$ and C_p (figure 7). Therefore the domain $\mathcal{G}(g(o), r)$ is removed from $K(D^*)$.

4.3.3 Adding existence constraints

Finally, note that γ will exist if and only if the two segments t_0t_1 and t_2n_3 exist (even with a null length). When C_p is located in the convex domain bounded by n_1n_2 and n_2n_3 (say, if C_p is 'outside' the turn Λ) then t_0t_1 always exists; otherwise t_1 must be located outside the circle C_p. In order to meet this constraint, the domains $\mathcal{G}(C_p, r)$ is removed from $K(D^*)$. t_2n_3 exists if the point n_3 is located outside the circle C. In order to satisfy meet this constraint, the domains $\mathcal{G}(n_3, r)$ is removed from $K(D^*)$.

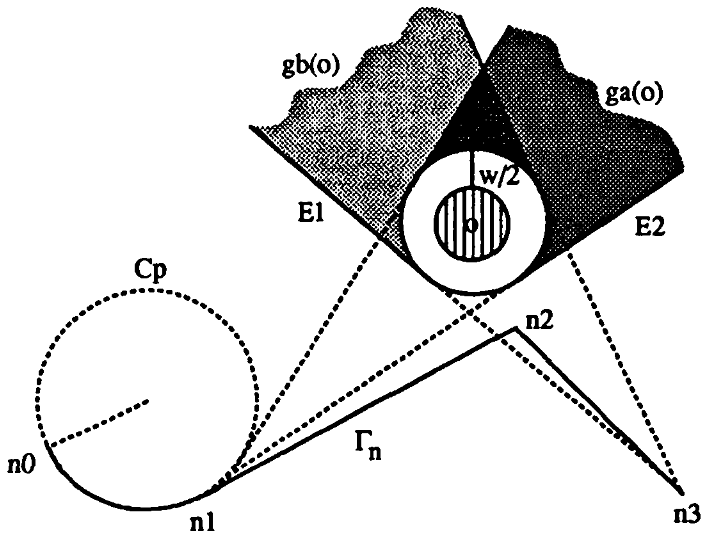

Figure 7: $g_a(o)$ and $g_b(o), o \subset D$

4.3.4 The final definition of CCS

Now, it is possible to define formally the curvature centres space CCS for the turn Λ:

$$\begin{aligned} CCS &= K(D^*) &-& \bigcup_{o \in \mathcal{O}, o \subset D} \{\mathcal{G}(f(o), r), \mathcal{G}(g(o), r)\} \\ & &-& \mathcal{G}(n_3, r) \\ & &-& \mathcal{G}(C_p, r) \text{ if } C_p \text{ 'outside' } \Lambda \end{aligned}$$

Figure 8 shows an example of curvature centres space for the turn ($\widehat{n_0 n_1}$, n_1n_2, n_2n_3) among three black striped circular obstacles. The vehicle considered along with its gyration radius is represented in the lower left window. The different domains to be removed from $K(D^*)$ are represented in the main window. The thick black line is the limit of $K(D^*)$ while the dotted area represents CCS for this particular turn. If the centre of the turning arc is picked up in CCS then the resulting trajectory Γ for the turn is collision-free and topologically equivalent to Γ_n. Let us notice that a sufficient condition for Γ to exist is that $CCS \neq \emptyset$.

4.4 Building the curvature centres space for the first and the last turn of S

The method presented above deals with a general turn—i.e. it is assumed that there exists a previous turn and a next one as well. Dealing with the first or the last turn is slightly different because these turns must meet an additional orientation constraint (see §3.1).

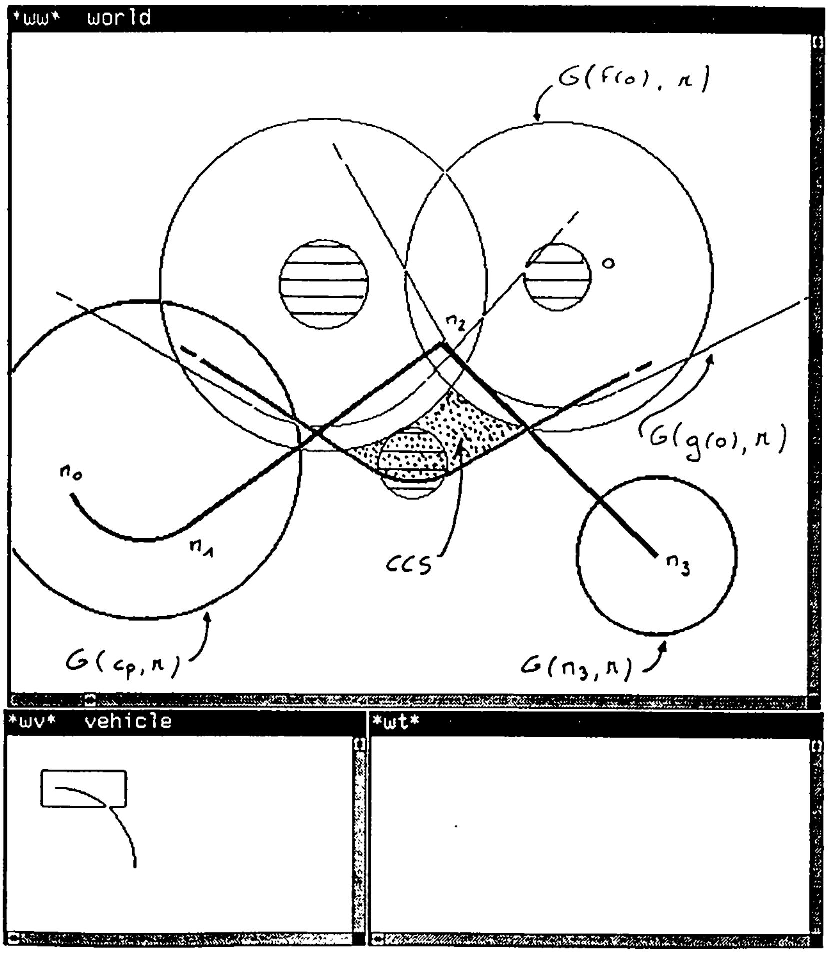

Figure 8: an example of curvature centres space

Let us consider the first turn case. A first turn is characterized by the t-uple $(n_1n_2,\ n_2n_3)$, there is no previous turn and M starts from the position n_1 with the orientation $\overrightarrow{n_1n_2}$. The method operates in two steps:

1. It generates a trajectory with the general method considering that $c_p = n_1$ and that $r_p = 0$. It provides us with a trajectory $\Gamma = (n_1t_1, \widehat{t_1t_2}, t_2n_3)$.

2. It substitutes the segment n_1t_1 for the sequence made up of the arc $\widehat{n_1t_0}$ of radius r tangential to n_1n_2 at point n_1 and the straight segment t_0t_1 connecting this arc to $\widehat{t_1t_2}$ (figure 9).

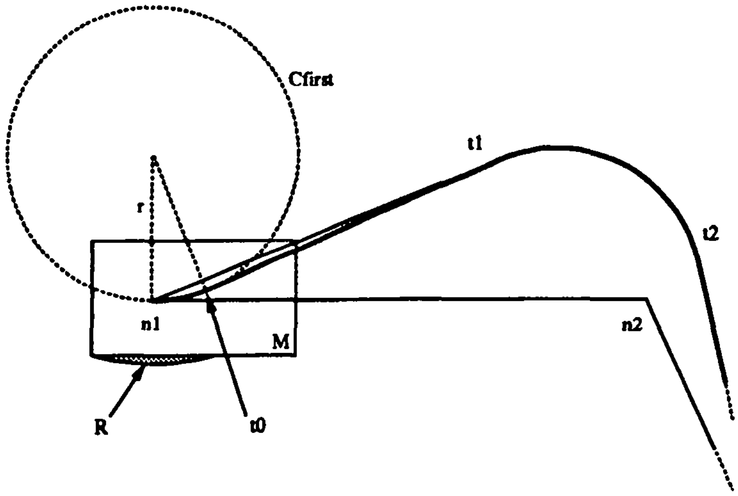

Figure 9: a first turn

One problem arises from this method: is the new sequence $(\widehat{n_1t_0},\ t_0t_1)$ collision-free? It is if we assume that the hypothesis (4b) of §3.1 is true. Let us consider figure 9, the hypothesis (4b) states that M is able to make a left-hand turn starting from n_1 without generating a collision with an obstacle located on the right side of the segment t_0t_1. In other words, there is no obstacle in the shaded region R. It can be easily shown that the new sequence $(\widehat{n_1t_0},\ t_0t_1)$ is collision-free with respect to the obstacles of $\mathcal{O}$.

Besides the problem of the existence of the arc $\widehat{n_1t_0}$ lead us to modify slightly the definition of the curvature centres space CCS for a first turn. Indeed, in the case illustrated by figure 9, the arc $\widehat{n_1t_0}$ exists if and only if $\widehat{t_1t_2}$ is located outside the circle C_{first}—i.e. the circle of radius r tangential to n_1n_2 at point n_1 and located in the concave domain bounded by n_1n_2 and n_2n_3. Therefore and in order to guarantee the existence of $\widehat{n_1t_0}$, the domain $\mathcal{G}(C_{first}, r)$ is removed from $K(D^*)$. The curvature centres space CCS for a first turn is then defined as:

$$
\begin{aligned}
CCS \;=\; K(D^*) \;&-\; \bigcup_{o\in\mathcal{O}, o\subset D}\{\mathcal{G}(f(o),r), \mathcal{G}(g(o),r)\}\\
&-\; \mathcal{G}(n_3, r)\\
&-\; \mathcal{G}(C_{first}, r)
\end{aligned}
$$

Dealing with a last turn is done in a similar fashion. The same two-step method is applied. Hypothesis (4c) of §3.1 guarantees that the new sequence is collision-free. The definition of the curvature centres space is adapted in the same way.

4.5 Searching the curvature centres space

In the previous sections, we have shown how to build the curvature centres space CCS for any turn Λ. we have also shown that a sufficient condition to generate a trajectory Γ satisfying all the constraints presented in §3.1 is that $c \in CCS$. Then the next step consists in exploring CCS in order to find such an appropriate centre c.

Let us notice that CCS is not necessarily connected and that its connected components are not necessarily simply connected (there may be holes). Therefore the full exploration of CCS is a very costly task (this point is emphasized in [9]). Hopefully this full exploration is not necessary to solve our problem. Indeed the vehicle is assumed to follow a trajectory somewhat close to the theoretical trajectory represented by Γ_n (remember that Γ_n is a subpart of the spine S which captures the structure of the world). Since the easiest way to perform Λ with a circular transition of radius r is to follow the arc tangential to the segments n_1n_2 and n_2n_3, it seems natural to expect c to be in the neighbourhood of the centre g of this particular arc. Therefore a good heuristic is to restrict the exploration of CCS to the disk C_g centered in g and whose radius r_g is a function of r (e.g. $r_g = 2r$).

In practice, the domain C_g is discretized and then explored. If the search fails then it is assumed that the vehicle cannot perform Λ with the given radius of gyration. Since CCS is not fully explored, The algorithm is not complete and so is liable to fail to find a solution for some turns.

Different heuristic strategies may be used to explore C_g depending on the turn Λ considered. In the current implementation, we have considered two different cases:

1. The default case. If g belongs to CCS then $c = g$. Otherwise C_g is explored using circles centered in g and of growing radius (figure 10).

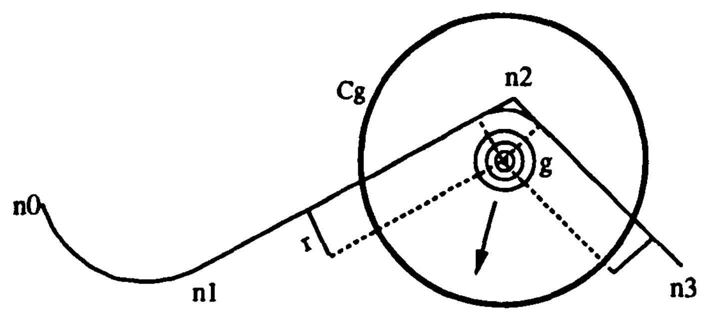

Figure 10: the default case

2. The 'close' turn case. This heuristic is applied when the turn following Λ is 'close'. The next turn is said to be close if g belongs to the half plane P that contains n_3 and whose boundary is the bissector of the segments n_1n_2 and n_3n_4 (figure 11). A U-turn is a typical close turn case. In this case, the previous strategy is not appropriate because it generally produces a turning arc for Λ which overconstrains the next turn and forbid to find a solution for it. To get round this problem, we make use of an alternative strategy: the subpart of C_g outside P is explored first along lines parallel to the boundary of P. Eventually the subpart of C_g inside P is explored in the same way.

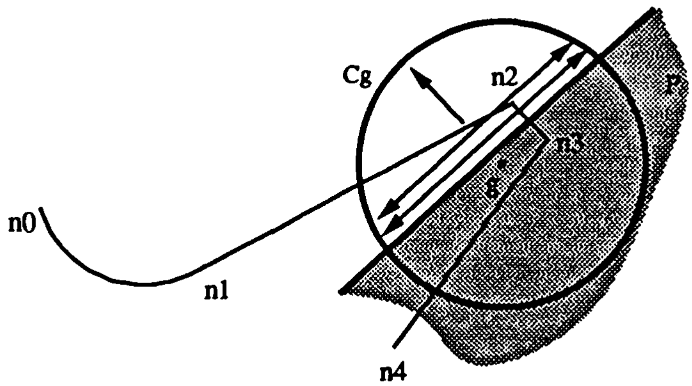

Figure 11: the 'close' turn case

Note: when two or more consecutive right-hand (resp. left-hand) turns are close. It is possible to extend the concept of turn as defined earlier in §4.1 in order to consider simultaneously the consecutive turns and to merge them into one single turn to be executed through one single turning arc. For example, let us consider the U-turn case depicted in figure 12. The current turn Λ is characterized by the t-uple $(\widehat{n_0n_1}, n_1n_2, n_2n_3)$. So far the next turn Λ_n is characterized by the two segments n_1n_2 and n_3n_4. Λ and Λ_n are both right-hand turns and Λ_n is close to Λ. In this case, Λ and Λ_n can be merged into one new turn Λ^+ characterized by the t-uple $(\widehat{n_0n_1}, n_1n_2, n_2n_3, n_3n_4)$. The definition of CCS for Λ^+ remains the same and the trajectory Γ^+ searched for is a connected sequence $(\widehat{n_0t_0}, t_0t_1, \widehat{t_1t_2}, t_2n_4)$.

5 Experiments

A prototype of the trajectory planner was implemented in Lucid Common Lisp on a 3/60 Sun workstation. The algorithm was successfully tested on several

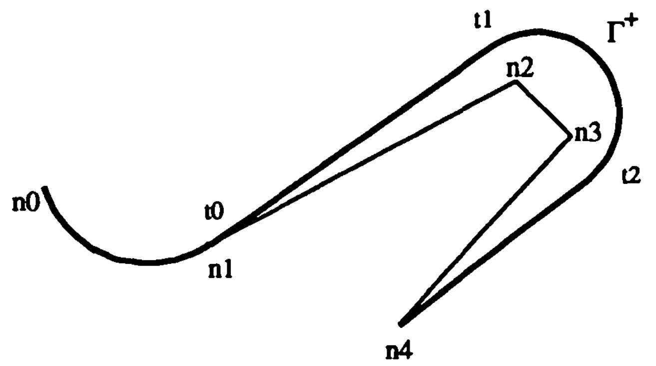

Figure 12: merging turns

examples of subsets of the road network (mainly intersections) and on more complex environments (figure 14 and 15).

These experiments have shown that the algorithm works fairly well as long as the environment is not too tricky. This is obviously the case when one deals with subsets of the road network since this environment is especially designed for car-like vehicles.

These experiments have also shown that the set of solutions for a particular turn is generally limited. In other words, the curvature centres space CCS is a very restricted area. However it appears that a significant part of this area is generally located in the neighbourhood of g. This validates the general turn strategy used to explore CCS. Figure 13 illustrates this: the left hand screen shows the environment, the spine $\mathcal{S}$ of the lane that the vehicle has to follow and the resulting trajectory computed by the algorithm. The striped areas are obstacles. The thin polygonal line represents $\mathcal{S}$ while the thick line is the final trajectory. The radii of the circular arcs of the trajectory are displayed). The right-hand screen shows the curvature centres space for the second turn of $\mathcal{S}$. CCS is represented by the black area.

Figures 14 and 15 show several results of the trajectory planner. The left-hand part of figure 14 illustrates the U-turn case. The algorithm is able to find a trajectory in this much constrained situation thanks to the 'close' turn strategy used to explore CCS. The situation of the right-hand part of figure 14 corresponds to a parking trajectory (without manœuvres). This is also a very constrained situation.

As for the complexity issues, the overall algorithmic complexity is $\mathcal{O}(nm)$ where n is the number of obstacles in the world and m the number of segments in the spine $\mathcal{S}$. The use of heuristic strategies to explore CCS spares us the cost of having to represent CCS explicitly and explore it thoroughly.

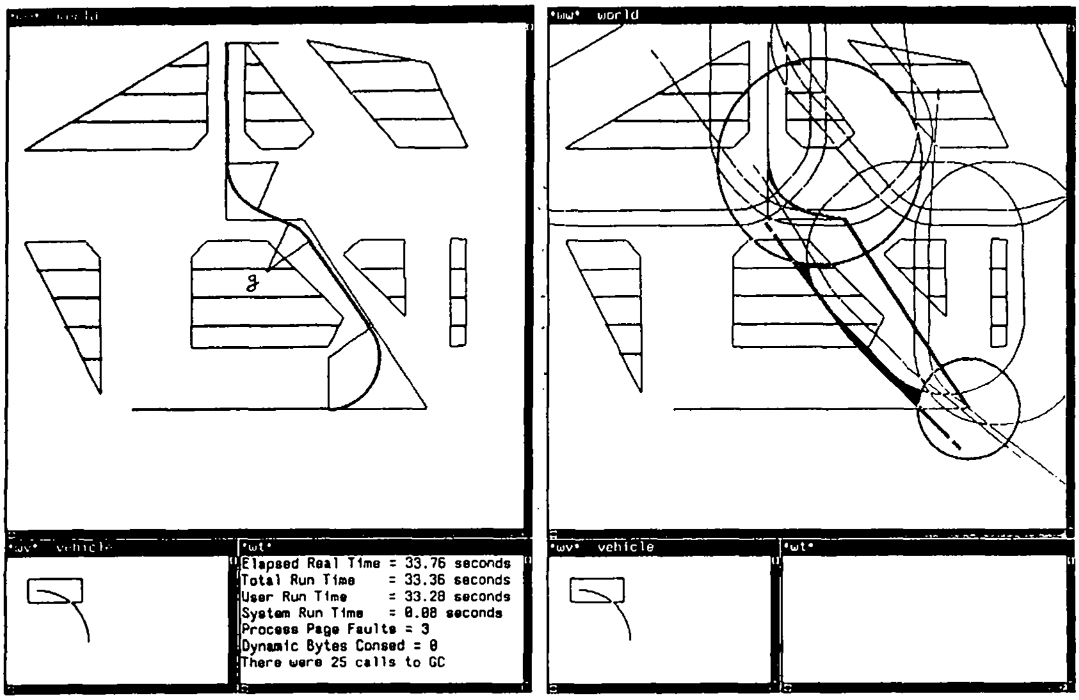

Figure 13:

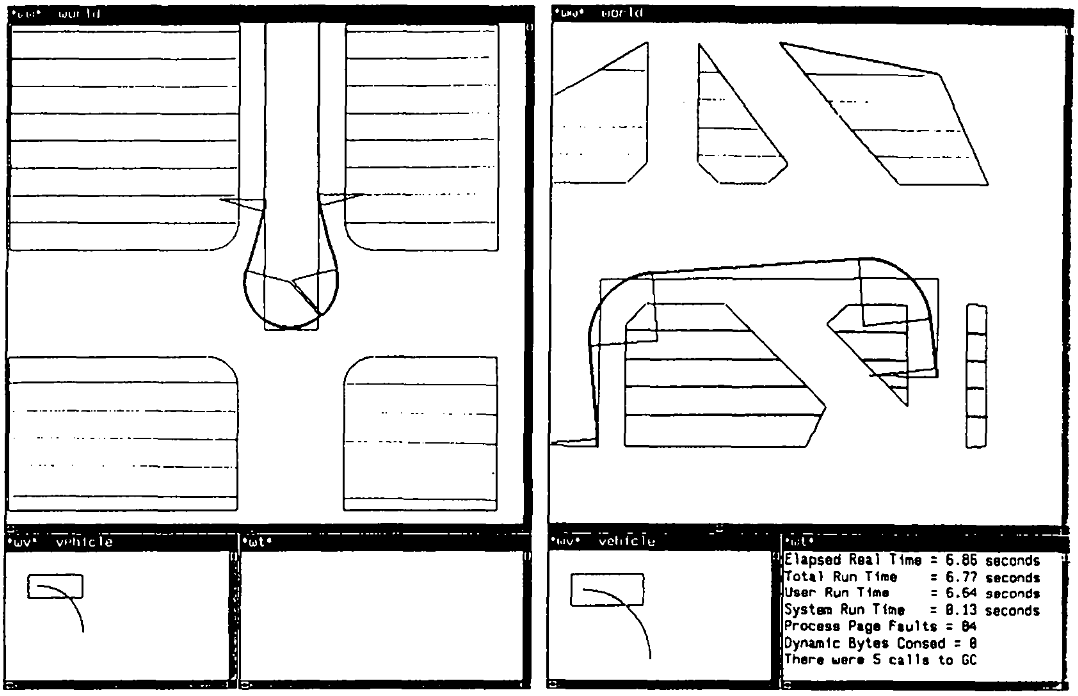

Figure 14:

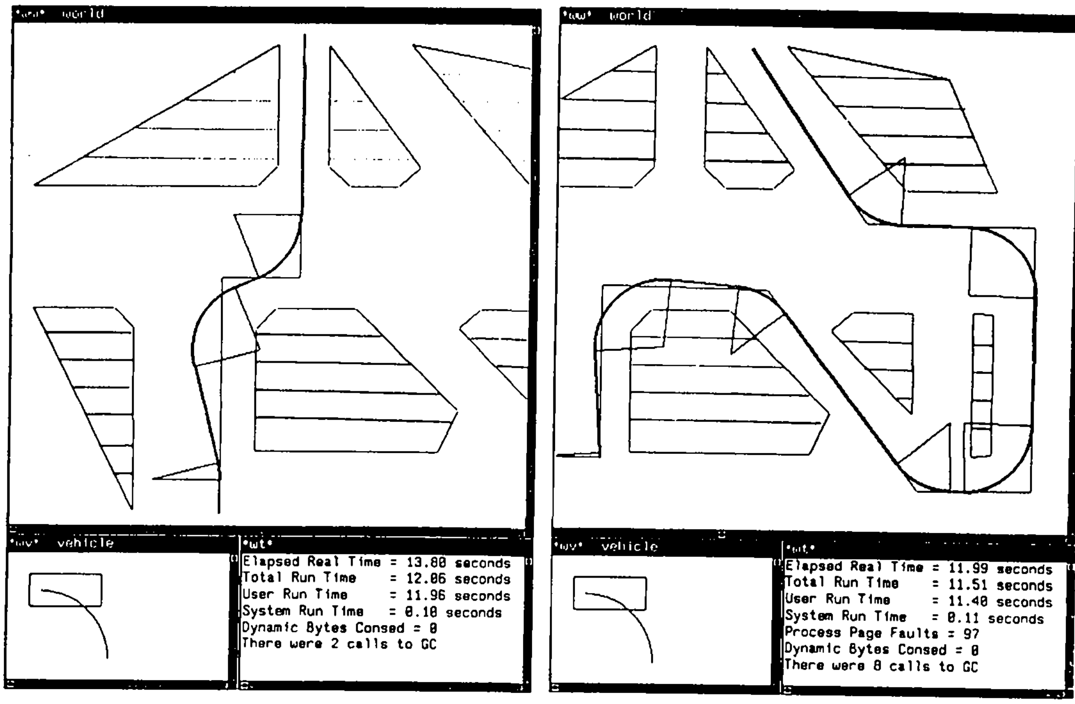

Figure 15:

6 Conclusion

In this report, we have presented an algorithm aimed at planning smooth trajectories for a car-like vehicle—i.e. a non-holonomic vehicle whose turning radius is lower bounded—in a static and structured world. As for the structure of the world, we assume the existence of natural lanes within which the vehicle is able to move. The planning algorithm, when given the polygonal line $\mathcal{S}$ representing the spine of the lane that the vehicle has to follow, generates a trajectory $\mathcal{C}$ verifying the following properties: (1) $\mathcal{C}$ is collision-free, (2) $\mathcal{C}$ is smooth—i.e. without backing up manœuvres, (3) $\mathcal{C}$ is executable by the vehicle according to its own kinematic constraints and (4) $\mathcal{C}$ is 'topologically equivalent' to $\mathcal{S}$; in other words, $\mathcal{C}$ must remain in the lane defined by $\mathcal{S}$. The generated trajectory $\mathcal{C}$ is made up of straight segments and tangential circular arcs of a given radius r. The basic principle of the algorithm is to search for the centre of each circular arc in a particular domain called 'curvature centres space' [9]. The characteristics of the problem to be solved enable our planning algorithm to operate within the workspace of the vehicle rather than within its configuration space. This accounts for the efficiency of our planner. However this efficiency is obtained at the expense of completeness because the algorithm makes use of a heuristic which is not complete. But tests in roadway-like environments have proved successful.

Acknowledgments

The author would like to thank J-P. Laumond for his helpful comments. This work was supported by the French Ministry of Research and Technology and the European Prometheus Eureka project.

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] S. Fortune and G. Wilfong. Planning constrained motions. In *Proc. of the Symp. on Theory of Computer Science*, pages 445–459, Chicago, Il (USA), May 1988. ACM.

[3] Th. Fraichard and C. Laugier. Planning movements for several coordinated vehicles. In *Proc. of the Int. Workshop on Intelligent Robots and Systems*, pages 466–472, Tsukuba (Japan), Sep. 1989. IEEE/RSJ.

[4] Th. Fraichard, C. Laugier, and G. Liévin. Robot motion planning: the case of non-holonomic mobiles in a dynamic world. In *Proc. of the Int. Workshop on Intelligent Robots and Systems*, pages 757–764, Tsuchiura (Japan), July 1990. IEEE/RSJ.

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

[6] P. Jacobs, J-P. Laumond, M. Taix, and R. Murray. Fast and exact trajectory planning for mobile robots and other systems with non-holonomic constraints. Research Report 90318, Laboratoire d'Automatique et d'Analyse des Systèmes, Toulouse (Fr), Sep. 1990.

[7] J-C. Latombe. A fast path planner for a car-like indoor mobile robot. In *Proc. of the Nat. Conf. on Artificial Intelligence*, Anaheim, Ca (USA), July 1991. AAAI.

[8] J-P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. *Intelligent Autonomous Systems*, pages 346–354, Dec. 1986.

[9] J-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, pages 1120–1123, Milan (Italy), Aug. 1987.

[10] J-P. Laumond. Obstacle growing in a non-polygonal world. *Inf. Process. Lett.*, 25:41–50, Apr. 1987.

[11] J-P. Laumond. non-holonomic motion planning versus controllability via the multibody car-system example. Working paper STAN-CS 90-1345, Computer Science Dept, Stanford Univ, Ca (USA), 1990.

[12] J-P. Laumond, T. Siméon, R. Chatila, and G. Giralt. Trajectory planning and motion control for mobile robots. In J-D. Boissonnat and J-P. Laumond, editors, *Geometry and Robotics*, pages 133–149. Lecture Notes in Computer Science, Vol 391, Springer-Verlag, 1989.

[13] J-P. Laumond, M. Taix, and P. Jacobs. A motion planner for car-like robots based on a mixed global/local approach. In *Proc. of the Int. Workshop on Intelligent Robots and Systems*, pages 765–773, Tsuchiura (Japan), July 1990. IEEE/RSJ.

[14] Z. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Trans. on Robotics and Automation*, 6, 1990.

[15] M. Sharir. Algorithmic motion planning in robotics. Research Report 392, Courant Institute, New-York (USA), 1988.

[16] P. Tournassoud. *Géométrie et intelligence artificielle pour les robots.* Traité des nouvelles technologies, série robotique. Hermès, Paris (Fr), 1988. In French.

[17] P. Tournassoud and O. Jehl. Motion planning for a mobile robot with a kinematic constraint. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 1785–1790, Philadelphia, Pen (USA), Apr. 1988. IEEE.

[18] G. Wilfong. Motion planning for an autonomous vehicle. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 529–533, Philadelphia, Pen (USA), Apr. 1988.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399