Nonholonomic Motion Planning for Mobile Robots

Tutorial Notes

J.P. Laumond

LAAS-CNRS
7 Avenue du Colonel Roche
31077 Toulouse Cedex 4, France

jpl@laas.fr, http://www.laas.fr/~jpl

May 1998

Foreword

These notes are mainly based on the following previously published papers:

- J.P. Laumond, S. Sekhavat and F. Lamiraux, "Guidelines in nonholonomic motion planning" in *Robot Motion Planning and Control*, J.P. Laumond Ed., Lecture Notes in Control and Information Science, 229, Springer Verlag, 1998.
- M. Vendittelli and J.P. Laumond, "Visible positions for a car-like robot amidst obstacles," *Algorithms for Robotic Motion and Manipulation*, J.P. Laumond and M. Overmars Eds, A.K. Peters, 1997.
- J.P. Laumond, P. Jacobs, M. Taïx and R. Murray, "A motion planner for nonholonomic mobile robot," *IEEE Trans. on Robotics and Automation*, Vol. 10, 1994.
- J.P. Laumond, " Singularities and topological aspects in nonholonomic motion planning," *Nonholonomic Motion Planning*, Zexiang Li and J.F. Canny Eds, The Kluwer International Series in Engineering and Computer Science 192, 1992.
- F. Lamiraux, S. Sekhavat and J.P. Laumond, "Motion planning and control for Hilare pulling a trailer," (to be published)

The work about mobile robots with trailer benefits from Ph.D. theses of S. Sekhavat (LAAS-CNRS Report 96505, Toulouse, November 1996) and F. Lamiraux (LAAS-CNRS Report 97327, Toulouse, September 1997). The work about shortest paths is based on joined work with M. Vendittelli and C. Nissoux.

Contents

1 Introduction

Mobile robots did not wait to know that they were nonholonomic to plan and execute their motions autonomously. It is interesting to notice that the first navigation systems have been published in the very first International Joint Conferences on Artificial Intelligence from the end of the 60's. These systems were based on seminal ideas which have been very fruitful in the development of robot motion planning: as examples, in 1969, the mobile robot Shakey used a grid-based approach to model and explore its environment [82]; in 1977 Jason used a visibility graph built from the corners of the obstacles [121]; in 1979 Hilare decomposed its environment into collision-free convex cells [41].

At the end of the 70's the studies of robot manipulators popularized the notion of configuration space of a mechanical system [72]; in this space the "piano" becomes a point. The motion planning for a mechanical system is reduced to path finding for a point in the configuration space. The way was open to extend the seminal ideas and to develop new and well-grounded algorithms (see Latombe's book [59]).

One more decade, and the notion of nonholonomy (also borrowed from Mechanics) appears in the literature [61] on robot motion planning through the problem of car parking which was not solved by the pioneering mobile robot navigation systems. Nonholonomic Motion Planning then becomes an attractive research field [71].

These lecture notes give an account of the recent developments of the research in this area by focusing on its application to mobile robots.

Nonholonomic systems are characterized by constraint equations involving the time derivatives of the system configuration variables. These equations are non integrable; they typically arise when the system has less controls than configuration variables. For instance a car-like robot has two controls (linear and angular velocities) while it moves in a 3-dimensional configuration space. As a consequence, any path in the configuration space does not necessarily correspond to a feasible path for the system. This is basically why the purely geometric techniques developed in motion planning for holonomic systems do not apply directly to nonholonomic ones.

While the constraints due to the obstacles are expressed directly in the manifold of configurations, nonholonomic constraints deal with the tangent space. In the presence of a link between the robot parameters and their derivatives, the first question to be addressed is: does such a link reduce the *accessible* configuration space ? This question may be answered by studying the structure of the distribution spanned by the Lie algebra of the system controls.

Now, even in the absence of obstacle, planning nonholonomic motions is not an easy task. Today there is no general algorithm to plan motions for any nonholonomic system so that the system is guaranteed to exactly reach a given goal. The only existing results are for approximate methods (which guarantee only that the system reaches a neighborhood of the goal) or exact methods for special classes of systems; fortunately, these classes cover almost all the existing mobile robots.

Obstacle avoidance adds a second level of difficulty. At this level we should take into account both the constraints due to the obstacles (i.e., dealing with the configuration parameters of the system) and the nonholonomic constraints linking the parameter derivatives. It appears necessary to combine geometric techniques addressing the obstacle avoidance together with control theory techniques addressing the special structure of the nonholonomic motions. Such a combination is possible through topological arguments.

Section 2 introduces the concept of controllability and shows critical issues in mobile robot systems modeling. Section 3 develops the relationship between obstacle avoidance and nonholonomy. Then we review various techniques developed in Control Theory allowing to steer a nonholonomic system from a configuration to another in the absence of obstacles (Section4). The core of the presentation is Section 5: we show how to integrate steering methods in general nonholonomic path planning schemes. Various techniques are presented and compared. Finally Section 6 overviews special approaches.

Sections 2 to 7 are self-contained. Technical developments are reported within the annexes. The first one deals with combinatorial issues of the Lie algebra machinery. Two proofs of controllability for a car-like robot enlights in a practical way the concept of Lie bracket in Annex B. Annex C presents efficient algorithms for car-like robot distance computation. Annex D develops a technical point about a steering method using sinusoidal inputs. All these materials deal with nonholonomic path planning which may be consider as the problem of planning *open loop* controls; the problem of the motion control is sketched in Annex E via the presentation of real experiments conduced on the mobile robot Hilare pulling a trailer.

2 Controllabilities of mobile robots

The goal of this section is to state precisely what kind of controllability and what level of mobile robot modeling are concerned by motion planning.

2.1 Controllabilities

Let us consider a n-dimensional manifold, $\mathcal{U}$ a class of functions of time t taking their values in some compact sub-domain $\mathcal{K}$ of $\mathbf{R}^m$. The control systems Σ considered in this chapter are differential systems such that

$$\dot{X} = f(X)u + g(X).$$

u is the control of the system. The $i-$th column of the matrix $f(X)$ is a vector field denoted by f_i. $g(X)$ is called the drift. An *admissible* trajectory is a solution of the differential system with given initial and final conditions and u belonging to $\mathcal{U}$.

The following definitions use Sussmann's terminology [114].

Definition 1 *Σ is locally controllable from X if the set of points reachable from X by an admissible trajectory contains a neighborhood of X. It is small-time controllable from X if the set of points reachable from X before a given time T contains a neighborhood of X for any T.*

A control system will be said to be small-time controllable if it is small-time controllable from everywhere.

Small-time controllability clearly implies local controllability. The converse is false.

Checking the controllability properties of a system requires the analysis of the control Lie algebra associated with the system. Considering two vector fields f and g, the Lie bracket $[f, g]$ is defined as being the vector field $\partial f.g - \partial g.f$ [1]. The following theorem (see [113]) gives a powerful result for symmetric systems (i.e., $\mathcal{K}$ is symmetric with respect to the origin) without drift (i.e, $g(X) = 0$).

Theorem 1 *A symmetric system without drift is small-time controllable from X iff the rank of the vector space spanned by the family of vector fields f_i together with all their brackets is n at X.*

Checking the Lie algebra rank condition (LARC) on a control system consists in trying to build a basis of the tangent space from a basis (e.g., a P. Hall family) of the free Lie algebra spanned by the control vector fields. Annex A develops the combinatorial aspects of such a machinery.

2.2 Mobile robots: from dynamics to kinematics

Modeling mobile robots with wheels as control systems may be addressed with a differential geometric point of view by considering only the classical hypothesis of "rolling without slipping". Such a modeling provides directly kinematic models of the robots. Nevertheless, the complete chain from motion planning to motion execution requires to consider the ultimate controls that should be applied to the true system. With this point of view, the kinematic model should be derived from the dynamic one. Both view points

[1] The k-th coordinate of $[f, g]$ is

$$[f,g][k] = \sum_{i=1}^{n} (g[i]\frac{\partial}{\partial x_i} f[k] - f[i]\frac{\partial}{\partial x_i} g[k]).$$

converge to the same modeling (e.g., [66]) but the later enlightens on practical issues more clearly than the former.

Let us consider two systems: a two-driving wheel mobile robot and a car (in [23] other mechanical structures of mobile robots are considered).

2.2.1 Two-driving wheel mobile robots

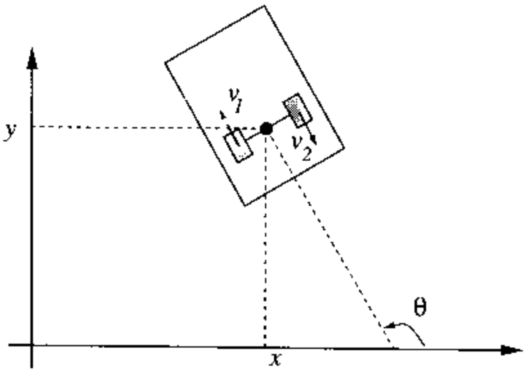


Figure 1: Two-driving wheel mobile robot

A classical locomotion system for mobile robot is constituted by two parallel driving wheels, the acceleration of each being controlled by an independent motor (Figure 1). The stability of the platform is insured by castors. The reference point of the robot is the midpoint of the two wheels; its coordinates, with respect to a fixed frame are denoted by (x, y); the main direction of the vehicle is the direction θ of the driving wheels. With ℓ designating the distance between the driving wheels the dynamic model is:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v}_1 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(v_1 + v_2)\cos\theta \\ \frac{1}{2}(v_1 + v_2)\sin\theta \\ \frac{1}{\ell}(v_1 - v_2) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \qquad (1)$$

with $|u_1| \leq u_{1,max}$, $|u_2| \leq u_{2,max}$ and v_1 and v_2 as the respective wheel speeds. Of course v_1 and v_2 are also bounded; these bounds appear at this level as "obstacles" to avoid in the 5-dimensional manifold. This 5-dimensional system is not small-time controllable from any point (this is due to the presence of the drift and to the bounds on u_1 and u_2).

By setting $v = \frac{1}{2}(v_1 + v_2)$ and $\omega = \frac{1}{\ell}(v_1 - v_2)$ we get the kinematic model which is expressed as the following 3-dimensional system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \qquad (2)$$

The bounds on v_1 and v_2 induce bounds v_{max} and ω_{max} on the new controls v and ω. Notice that v and ω should be C^1. The system is symmetric without drift; applying the LARC condition shows that it is small-time controllable from everywhere. A direct geometric proof appears in Annex B together with a second proof illustrating the geometric meaning of the Lie bracket.

2.2.2 Car-like robots

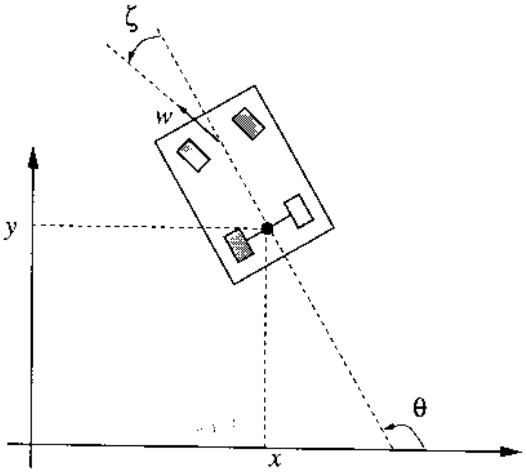


Figure 2: Car-like robot

From the driver's point of view, a car has two controls: the accelerator and the steering wheel. The reference point with coordinates (x, y) is the midpoint of the rear wheels (Figure 2). We assume that the distance between both rear and front axles is 1. We denote w as the speed of the front wheels of the car and ζ as the angle between the front wheels and the main direction θ of the car[2]. Moreover a mechanical constraint imposes $|\zeta| \leq \zeta_{max}$ and consequently a minimum turning radius. Simple computation shows that the dynamic model of the car is:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{w} \\ \dot{\zeta} \end{pmatrix} = \begin{pmatrix} w \cos\zeta \cos\theta \\ w \cos\zeta \sin\theta \\ w \sin\zeta \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \qquad (3)$$

with $|u_1| \leq u_{1,max}$ and $|u_2| \leq u_{2,max}$. This 5-dimensional system is not small-time controllable from everywhere.

[2] More precisely, the front wheels are not exactly parallel; we use the average of their angles as the turning angle.

A first simplification consists in considering w as a control; it gives a 4-dimensional system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\zeta} \end{pmatrix} = \begin{pmatrix} \cos\zeta\cos\theta \\ \cos\zeta\sin\theta \\ \sin\zeta \\ 0 \end{pmatrix} w + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \tag{4}$$

This new system is symmetric without drift; applying the LARC condition shows that it is small-time controllable from everywhere. Notice that w should be C^1. Up to some coordinate changes, we may show that this system is equivalent to the kinematic model of a two-driving wheel mobile robot pulling a "trailer" which is the rear axle of the car (see below). The mechanical constraint $|\zeta| \leq \zeta_{max} \leq \frac{\pi}{2}$ appears as an "obstacle" in $\mathbf{R}^2 \times (S^1)^2$.

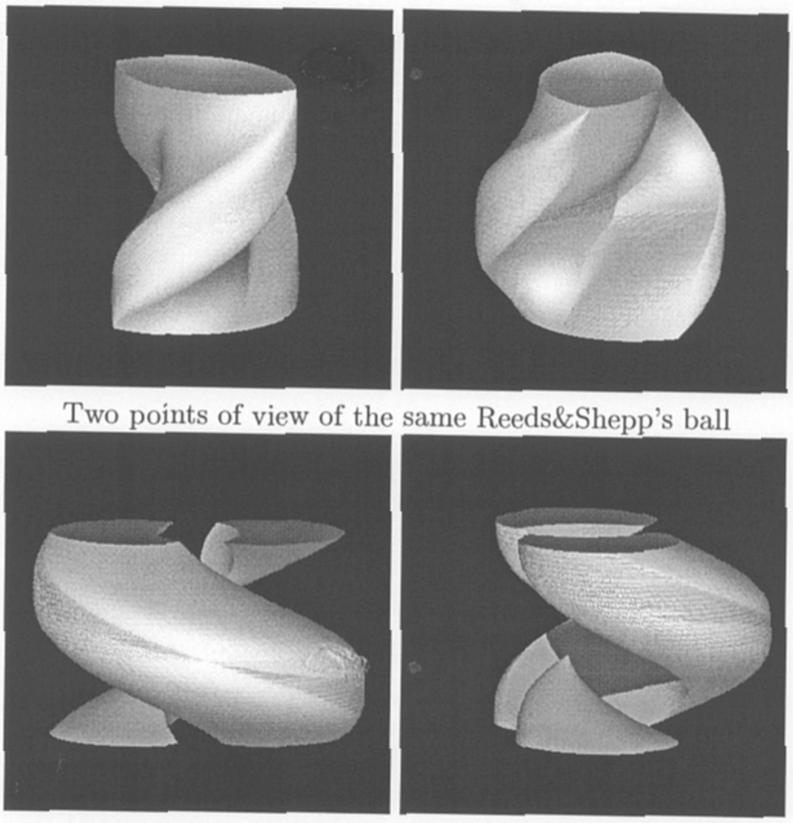
Let us assume that we do not care about the direction of the front wheels. We may still simplify the model. By setting $v = w\cos\zeta$ and $\omega = w\sin\zeta$ we get a 3-dimensionated control system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \tag{5}$$

By construction v and ω are C^1 and their values are bounded. This system looks like the kinematic model of the two-driving wheel mobile robot. The main difference lies on the admissible control domains. Here the constraints on v and ω are no longer independent. Indeed, by setting $w_{max} = \sqrt{2}$ and $\zeta_{max} = \frac{\pi}{4}$ we get: $0 \leq |\omega| \leq |v| \leq 1$. This means that the admissible control domain is no longer convex. It remains symmetric; we can still apply the LARC condition to prove that this system is small-time controllable from everywhere. The main difference with the two-driving wheel mobile robot is that the feasible paths of the car should have a curvature lesser than 1.

A last simplification consists in putting $|v| \equiv 1$ and even $v \equiv 1$; by reference to the work in [87] and [33] on the shortest paths in the plane with bounded curvature such systems will be called Reeds&Shepp's car and Dubins' car respectively (see Annex C for an overview of recent results on shortest paths for car-like robots). The admissible control domain of Reeds&Shepp's car is symmetric; LARC condition shows that it is small-time controllable from everywhere. Dubins' car is a system with drift; it is locally controllable but not small-time controllable from everywhere; for instance, to go from $(0, 0, 0)$ to $(1 - \cos\epsilon, \sin\epsilon, 0)$ with Dubins car takes at least $2\pi - \epsilon$ unity of time.

The difference between the small-time local controllability of Reeds&Shepp's car and the local controllability of Dubins' car may be illustrated geometrically. Figure 3 shows the accessibility surfaces in $\mathbf{R}^2 \times S^1$ of both systems for a fixed length of the shortest paths. Such surfaces have been computed from the synthesis of the shortest paths for these systems (see [106, 70, 20]). In the case of Reeds&Shepp's car, the surface encloses a neighborhood of the origin; in the case of Dubins' car the surface is not connected and it does not enclose any neighborhood of the origin.



Two points of view of the same Reeds&Shepp's ball

Two points of view of the same Dubins' "ball"

Figure 3: Accessibility domains by shortest paths of fixed length

2.3 Kinematic model of mobile robots with trailers

Let us now introduce the mobile robot with trailers which has been the canonical example of the work in nonholonomic motion planning; it will be the leading thread of the rest of the presentation.

Figure 4 (left) shows a two-driving wheel mobile robot pulling two trailers; each trailer is hooked up at the middle point of the rear wheels of the previous one. The distance between the reference points of the trailers is assumed to be 1. The kinematic model is defined by the following control system (see [66]) :

$$\dot{X} \;=\; f_1(X)v \;+ f_2(X)\omega \tag{6}$$

with

$X = (x, y, \theta, \varphi_1, \varphi_2)^T$
$f_1(X) = (\cos\theta,\ \sin\theta,\ 0,\ -\sin\varphi_1,\ \sin\varphi_1 - \cos\varphi_1 \sin\varphi_2)^T$ and
$f_2(X) = (0,\ 0,\ 1,\ 1,\ 0)^T$

Note that the first body can be viewed as the front wheels of a car; the system then appears as modeling a car-like robot pulling a trailer.

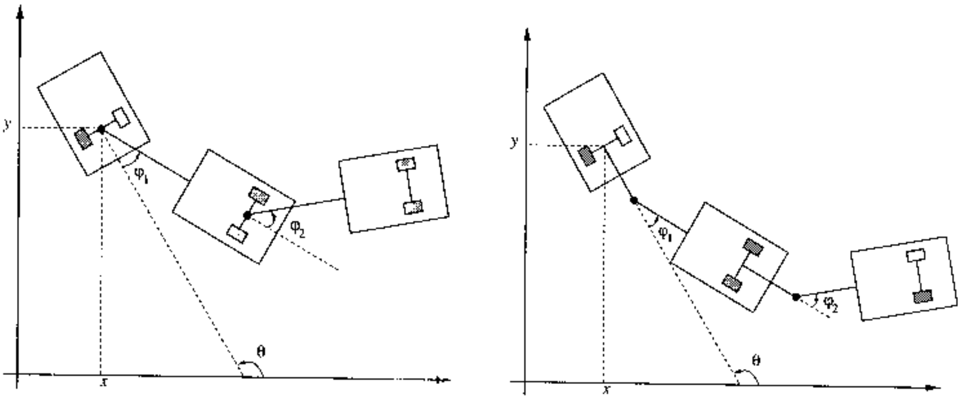


Figure 4: Two types of mobile robots with trailers.

After noticing that $[f_2, [f_1, f_2]] = f_1$, one may check that the family $\{f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]], [f_1, [f_1, [f_1, f_2]]]\}$ spans the tangent space at every point in $\mathbf{R}^2 \times (S^1)^3$ verifying $\varphi_1 \not\equiv \frac{\pi}{2}$ (regular points). The family $\{f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]], [f_1, [f_1, [f_1, [f_1, f_2]]]]\}$ spans the tangent space elsewhere (i.e., at singular points). Thanks to the LARC, we conclude that the system is small-time controllable at any point. Its degree of nonholonomy[3] is 4 at regular points and 5 at singular points. A more general proof of small-time controllability for this system with n trailers appears in [66].

Another hooking system is illustrated in Figure 4 (right). Let us assume that the distance between the middle point of the wheels of a trailer and the hookup of the preceding one is 1. The control system is the same as (6), with
$f_1(X) = (\cos\theta,\ \sin\theta,\ 0,\ -\sin\varphi_1,\ -\sin\varphi_2 \cos\varphi_1 + \cos\varphi_2 \sin\varphi_1 + \sin\varphi_1)^T$ and
$f_2(X) = (0,\ 0,\ 1,\ -1 - \cos\varphi_1,\ \sin\varphi_1 \sin\varphi_2 + \cos\varphi_1 \cos\varphi_2 + \cos\varphi_1)^T$

The family $\{f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]], [f_2, [f_1, f_2]]\}$ spans the tangent space at every point in $\mathbf{R}^2 \times (S^1)^3$ verifying $\varphi_1 \not\equiv \pi$, $\varphi_2 \not\equiv \pi$ and $\varphi_1 \not\equiv \varphi_2$ (regular points). The degree of nonholonomy is then 3 at regular points. The family $\{f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]], [f_1, [f_1, [f_1, f_2]]]\}$ spans the tangent space at points verifying $\varphi_1 \equiv \varphi_2$. The degree of nonholonomy at these points is then 4. When $\varphi_1 \equiv \pi$ or $\varphi_2 \equiv \pi$ the system is no more controllable; this is a special case of mechanical singularities.

[3] The minimal length of the Lie bracket required to span the tangent space at a point is said to be the degree of nonholonomy of the system at this point. The degree of nonholonomy of the system is the upper bound d of all the degrees of nonholonomy defined locally (see Annex A for details).

2.4 Admissible paths and trajectories

2.4.1 Constrained paths and trajectories

Let $\mathcal{CS}$ be the configuration space of some mobile robot (i.e., the minimal number of parameters locating the whole system in its environment). In the sequel a *trajectory* is a continuous function from some real interval $[0,T]$ in $\mathcal{CS}$. An *admissible* trajectory is a solution of the differential system corresponding to the kinematic model of the mobile robot (including the control constraints), with some initial and final given conditions. A *path* is the image of a trajectory in $\mathcal{CS}$. An *admissible* path is the image of an admissible trajectory.

The difference between the various kinematic models of the mobile robots considered in this presentation only concerns their control domains (Figure 5). It clearly appears that admissible paths for Dubins' car are admissible for Reeds&Shepp's car (the converse is false); admissible paths for Reeds&Shepp's car are admissible for the car-like robot (the converse is true); admissible paths for the car-like robot are admissible for the two-driving wheel mobile robot (the converse is false).

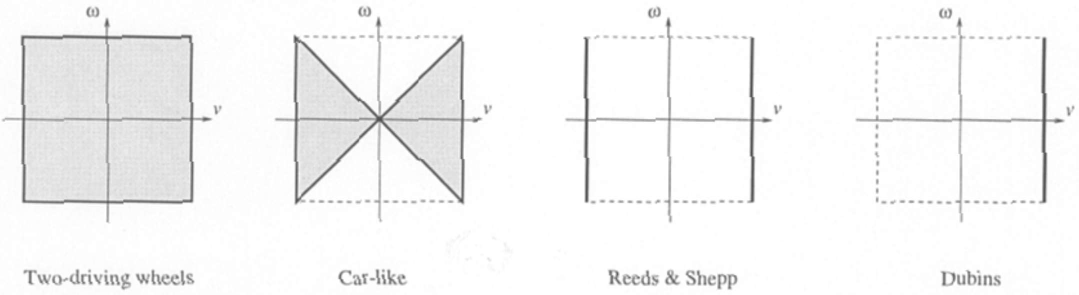


Figure 5: Kinematic mobile robot models: four types of control domains.

Remark 1: Due to the constraint $|\omega| < |v|$, the admissible paths for the car-like, Reeds&Shepp's and Dubins' robots have their curvature upper bounded by 1 everywhere. As a converse any curve with curvature upper bounded by 1 is an admissible path (i.e., it is possible to compute an admissible trajectory from it).

Remark 2: This geometric constraint can be taken into account by considering the four-dimensionated control system (4) with $|\zeta| \leq \frac{\pi}{4}$; the inequality constraint on the controls for the 3-dimensionated system is then transformed into a geometric constraint on the state variable ζ. Therefore the original control constraint $|\omega| < |v|$ arising in system (5) can be addressed by applying "obstacle" avoidance techniques to the system (4).

2.4.2 From paths to trajectories

The goal of nonholonomic motion planning is to provide *collision-free admissible paths* in the configuration space of the mobile robot system. Obstacle avoidance imposes a geometric point of view that dominates the various approaches addressing the problem. The motion planners compute paths which have to be transformed into trajectories.

In almost all applications, a black-box module allows to control directly the linear and angular velocities of the mobile robot. Velocities and accelerations are of course submitted bounds.

The more the kinematic model of the robot is simplified, the more the transformation of the path into a trajectory should be elaborated. Let us consider for instance an elementary path consisting of an arc of a circle followed by a tangent straight line segment. Due to the discontinuity of the curvature of the path at the tangent point, a two driving-wheel mobile robot should stop at this point; the resulting motion is clearly not satisfactory. This critical point may be overcome by "smoothing" the path before computing the trajectory. For instance clothoids and involutes of a circle are curves that account for the dynamic model of a two driving-wheel mobile robot: they correspond to bang-bang controls for the system (1) [47]; they may be used to smooth elementary paths [36].

Transforming an admissible path into an admissible trajectory is a classical problem which has been investigated in robotics community mainly through the study of manipulators (e.g., [90] for a survey of various approaches). Formal solutions exist (e.g., [104] for an approach using optimal control); they apply to our problem. Nevertheless, their practical programming tread on delicate numerical computations (see Annex E).

On the other hand, some approaches address simultaneously the geometric constraints of obstacle avoidance, the kinematic and the dynamic ones; this is the so-called "kinodynamic planning problem" (e.g., [30, 32, 88]). These methods consist in exploring the phase space (i.e., the tangent bundle associated to the configuration space of the system) by means of graph search and discretization techniques. In general, such algorithms provide approximated solutions (with the exception of one and two dimensional cases [83, 25]) and are time-consuming. Only few of them report results dealing with obstacle avoidance for nonholonomic mobile robots (e.g., [39]).

The following developments deal with nonholonomic *path* planning.

3 Path planning and small-time controllability

Path planning raises two problems: the first one addresses the *existence* of a collision-free admissible path (this is the decision problem) while the second one addresses the *computation* of such a path (this is the complete problem).

The results overviewed in this section show that the decision problem is solved for any small-time controllable system; even if approximated algorithms exist to solve the complete problem, the exact solutions deal only with some special classes of small-time controllable systems.

We may illustrate these statements with the mobile robot examples introduced in the previous section:

- Dubins' robot: this is the simplest example of a system which is locally controllable and not small-time controllable. For this system, the decision problem is solved when the robot is reduced to a point [38]. An approximated solution of the complete problem exists [46]; exact solutions exist for a special class of environments consisting of *moderated* obstacles (moderated obstacles are generalized polygons whose boundaries are admissible paths for Dubins' robot) [2, 16]. Notice that the decision problem is still open when the robot is a polygon.

- Reeds&Shepp's, car-like and two-driving wheel robots: these systems are small-time controllable. We will see below that exact solutions exist for both problems.

- Mobile robots with trailers: the two systems considered in the previous section are generic of the class of small-time controllable systems. For both of them the decision problem is solved. For the system appearing in Figure 4 (left) we will see that the complete problem is solved; it remains open for the system in Figure 4 (right).

Small-time controllability (Definition 1) has been introduced with a control theory perspective. To make this definition operational for path planning, we should translate it in purely geometric terms.

Let us consider a small-time controllable system, with $\mathcal{U}$ a class of control functions taking their values in some compact domain $\mathcal{K}$ of $\mathbf{R}^m$. We assume that the system is symmetric[4]. As a consequence, for any admissible path between two configurations X_1 and X_2, there are two types of admissible trajectories: the first ones go from X_1 to X_2, the second ones go from X_2 to X_1.

Let X be some given configuration. For a fixed time T, let $\mathcal{R}each_X(T)$ be the set of configurations reachable from X by an admissible trajectory before the time T. $\mathcal{K}$ being compact, $\mathcal{R}each_X(T)$ tends to $\{X\}$ when T tends to 0.

Because the system is small-time controllable, $\mathcal{R}each_X(T)$ contains a neighborhood of X. We assume that the configuration space is equipped with a (Riemannian) metric: any neighborhood of a point contains a ball centered at this point with a strictly positive radius. Then there exists a

[4] Notice that, with the exception of Dubins' robot, all the mobile robots introduced in the previous section are symmetric.

positive real number η such that the ball $B(X,\eta)$ centered at X with radius η is included in $\mathcal{R}each_X(T)$.

Now, let us consider a (not necessarily admissible) collision-free path γ with finite length linking two configurations X_{start} and X_{goal}. γ being compact, it is possible to define the clearance ϵ of the path as the minimum distance of γ to the obstacles[5]. ϵ is strictly positive. Then for any X on γ, there exists $T_X > 0$ such that $\mathcal{R}each_X(T_X)$ does not intersect any obstacle. Let η_{T_X} be the radius of the ball centered at X whose points are all reachable from X by admissible trajectories that do not escape $\mathcal{R}each_X(T_X)$. The set of all the balls $B(X,\eta_{T_X})$, $X \in \gamma$, constitutes a covering of γ. γ being compact, it is possible to get a finite sequence of configurations $(X_i)_{1\leq i\leq k}$ (with $X_1 = X_{start}$, $X_k = X_{goal}$), such that the balls $B(X_i,\eta_{T_{X_i}})$ cover γ.

Let us consider a point $Y_{i,i+1}$ lying on γ and in $B(X_i,\eta_{T_{X_i}})\cap B(X_{i+1},\eta_{T_{X_{i+1}}})$. Between X_i and $Y_{i,i+1}$ (respectively X_{i+1} and $Y_{i,i+1}$) there is an admissible trajectory (and then an admissible path) that does not escape $\mathcal{R}each_{X_i}(T_{X_i})$ (respectively $\mathcal{R}each_{X_{i+1}}(T_{X_{i+1}})$). Then there is an admissible path between X_i and X_{i+1} that does not escape $\mathcal{R}each_{X_i}(T_{X_i}) \cup \mathcal{R}each_{X_{i+1}}(T_{X_{i+1}})$; this path is then collision-free. The sequence $(X_i)_{1\leq i\leq k}$ is finite and we can conclude that there exists a collision-free admissible path between X_{start} and X_{goal}.

Theorem 2 *For symmetric small time controllable systems the existence of an admissible collision-free path between two given configurations is equivalent to the existence of any collision-free path between these configurations.*

Remark 3: We have tried to reduce the hypothesis required by the proof to a minimum. They are realistic for practical applications. For instance the compactness of $\mathcal{K}$ holds for all the mobile robots considered in this presentation. Moreover we assume that we are looking for admissible paths without contact with the obstacles: this hypothesis is realistic in mobile robotics (it does not hold any more for manipulation problems). On the other hand we suggest that two configurations belonging to the same connected component of the collision-free path can be linked by a finite length path; this hypothesis does not hold for any space (e.g., think to space with a fractal structure); nevertheless it holds for realistic workspaces where the obstacles are compact, where their shape is simple (e.g., semi-algebraic) and where their number is finite.

Consequence 1: Theorem 2 shows that the decision problem of motion planning for a symmetric small-time controllable nonholonomic system is the same as the decision problem for the holonomic associated one (i.e., when

[5] We consider that a configuration where the robot touches an obstacle is not collision-free.

the kinematics constraints are ignored): it is decidable. Notice that deciding whether some general symmetric system is small-time controllable (from everywhere) can be done by a only semi-decidable procedure (see Annex A). The combinatorial complexity of the problem is addressed in [108]. Explicit bounds of complexity have been recently provided for polynomial systems in the plane (see [91] and references therein).

Consequence 2: Theorem 2 suggests an approach to solve the complete problem. First, one may plan a collision-free path (by means of any standard methods applying to the classical piano mover problem); then, one approximates this first path by a finite sequence of admissible and collision-free ones. This idea is at the origin of a nonholonomic path planner which is presented below (Section 5.3). It requires effective procedures to steer a nonholonomic system from a configuration to another. The problem has been first attacked by ignoring the presence of obstacles (Section 4); numerous methods have been mainly developed within the control theory community; most of them account only for local controllability. Nevertheless, the planning scheme suggested by Theorem 2 requires steering methods that accounts for small-time controllability (i.e., not only for local controllability). In Section 5.1 we introduce a topological property which is required by steering methods in order to apply the planning scheme. We show that some among those presented in Section 4 verify this property, another one does not, and finally a third one may be extended to guaranty the property.

4 Steering methods

What we call a *steering method* is an algorithm that solves the path planning problem without taking into account the geometric constraints on the state. Even in the absence of obstacles, computing an admissible path between two configurations of a nonholonomic system is not an easy task. Today there is no algorithm that guarantees any nonholonomic system to reach an accessible goal exactly. In this section we present the main approaches which have been applied to mobile robotics.

4.1 From vector fields to effective paths

The concepts from differential geometry that we want to introduce here are thoroughly studied in [110, 123, 111, 112]. They give a combinatorial and geometric point of view of the path planning problem.

Choose a point X on a manifold and a vector field f defined around this point. There is exactly one path $\gamma(\tau)$ starting at this point and following f. That is, it satisfies $\gamma(0) = X$ and $\dot{\gamma}(\tau) = f(\gamma(\tau))$. One defines the exponential of f at point X to be the point $\gamma(1)$ denoted by $e^f.X$. Therefore

e^f appears as an operation on the manifold, meaning "slide from the given point along the vector field f for unit time". This is a diffeomorphism. With α being a real number, applying $e^{\alpha f}$ amounts to follow f for a time α. In the same way, applying e^{f+g} is equivalent to follow $f+g$ for unit time.

It remains that, whenever $[f,g] \neq 0$, following directly $\alpha f + \beta g$ or following first αf, then βg, are no longer equivalent. Intuitively, the bracket $[f,g]$ measures the variation of g along the paths of f; in some sense, the vector field g we follow in $\alpha f + \beta g$ is not the same as the vector field g we follow after having followed αf first (indeed g is not evaluated at the same points in both cases).

Assume that $f_1, \ldots, f_n$ are vector fields defined in a neighborhood $\mathcal{N}$ of a point X such that at each point of $\mathcal{N}$, $\{f_1, \ldots, f_n\}$ constitutes a basis of the tangent space. Then there is a smaller neighborhood of X on which the maps $(\alpha_1, \ldots, \alpha_n) \mapsto e^{\alpha_1 f_1 + \cdots + \alpha_n f_n} \cdot X$ and $(\alpha_1, \ldots, \alpha_n) \mapsto e^{\alpha_n f_n} \cdots e^{\alpha_1 f_1} \cdot X$ are two coordinate systems, called the first and the second normal coordinate system associated to $\{f_1, \ldots, f_n\}$.

The Campbell-Baker-Hausdorff-Dynkin formula states precisely the difference between the two systems: for a sufficiently small τ, one has:

$$e^{\tau f} \cdot e^{\tau g} = e^{\tau f + \tau g - \frac{1}{2}\tau^2 [f,g] + \tau^2 \epsilon(\tau)}$$

where $\epsilon(\tau) \to 0$ when $\tau \to 0$.

Actually, the whole formula as proved in [123] gives an explicit form for the ϵ function. More precisely, ϵ yields a formal series whose coefficients c_k of τ^k are combinations of brackets of degree k,[6] i.e.

$$\tau^2 \epsilon(\tau) = \sum_{k=3}^{\infty} \tau^k c_k$$

Roughly speaking, the Campbell-Baker-Hausdorff-Dynkin formula tells us how a small-time nonholonomic system can reach *any* point in a neighborhood of a starting point. This formula is the hard core of the local controllability concept (Annex B shows how to use this formula to prove the controllability of a car-like robot). The formula yields methods for *explicitly computing admissible paths* in a neighborhood of a point.

4.2 Nilpotent systems and nilpotentization

One method among the very first ones has been defined by Lafferiere and Sussmann [54] in the context of nilpotent system. A control system is nilpotent as soon as the Lie brackets of the control vector fields vanish from some given length (see Annex A for details).

For small-time controllable nilpotent systems it is possible to compute a basis $\mathcal{B}$ of the Control Lie Algebra $LA(\Delta)$ from a Philipp Hall family (see

[6] As an example the degree of $[[f,g],[f,[g,[f,g]]]]$ is 6.

Annex A). The method assumes that a holonomic path γ is given. If we express locally this path on $\mathcal{B}$, i.e., if we write the tangent vector $\dot{\gamma}(t)$ as a linear combination of vectors in $\mathcal{B}(\gamma(t))$, the resulting coefficients define a control that steers the holonomic system along γ. Because the system is nilpotent, each exponential of Lie bracket can be developed *exactly* as a finite combination of the control vector fields: such an operation can be done by using the Campbell-Baker-Hausdorff-Dynkin formula above. It is then possible to compute an admissible and *piecewise constant* control u for the nonholonomic system that steers the system *exactly* to the goal.

For a general system, Lafferiere and Sussmann reason as if the system were nilpotent of order k. In this case, the synthesized path deviates from the goal. Nevertheless, thanks to a topological property, the basic method may be used in an iterated algorithm that produces a path ending as close to the goal as wanted.

In [45], Jacob gives an account of Lafferiere and Sussmann's strategy by using another coordinate system. This system is built from a Lyndon basis of the free Lie algebra [127] instead of a P. Hall basis. This choice reduces the number of pieces of the solution.

In [11], Bellaïche *et al* apply the nilpotentization techniques developed in [10] (see also [43]). They show how to transform any controllable system into a canonical form corresponding to a nilpotent system approximating the original one. Its special triangular form allows to apply sinusoidal inputs (see below) to steer the system locally. Moreover, it is possible to derive from the proposed canonical form an estimation of the metrics induced by the shortest feasible paths. This estimation holds at regular points (as in [126]) as well as at singular points. These results are critical to evaluate the combinatorial complexity of the approximation of holonomic paths by a sequence of admissible ones (see Section 5.7).

The mobile robots considered in this presentation are not nilpotent[7]. A nilpotentization of this system appear in [54]. Nilpotentization techniques do not usually deal with the singularities of the system. Accounting singularities has been recently addressed in [125] for the case of a mobile robot pulling two trailers.

We conclude this section by the nilpotentization of a mobile robot pulling a trailer [11].

[7] Consider the system (2); let us denote f_1 and f_2 the two vector fields corresponding to a straight line motion and a rotation respectively. By setting $ad_f(g) = [f, g]$, we check that $ad_{f_2}^{2m}(f_1) = (-1)^m f_1 \neq 0$.

Example: Let us consider the control system 6:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\sin\varphi \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} u_2$$
$$= \quad f_1 u_1 \quad + \quad f_2 u_2$$

where (x, y) defines the position of the mobile robot, θ its direction and φ the angle of the trailer with respect to the mobile robot.

The coordinates of vector fields $f_3 = [f_1, f_2]$ and $f_4 = [f_1, [f_1, f_2]]$ are respectively:

$$f_3 = \begin{pmatrix} \sin\theta \\ -\cos\theta \\ 0 \\ \cos\varphi \end{pmatrix} \qquad f_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

We check easily that $\{f_1, f_2, f_3, f_4\}$ is a basis of the tangent space at every point of the manifold $\mathbf{R}^2 \times (S^1)^2$. Let $X_0 = (x_0, y_0, \theta_0, \varphi_0)$ and $X = (x, y, \theta, \varphi)$ be two points of the manifold. By writing $\Delta x = x - x_0$, $\Delta y = y - y_0$, $\Delta\theta = \theta - \theta_0$ and $\Delta\varphi = \varphi - \varphi_0$, the coordinates (y_1, y_2, y_3, y_4) of X in the chart attached to X_0 with the basis $\{f_1, f_2, f_3, f_4\}(X_0)$ are:

$$\begin{aligned} y_1 &= \cos\theta_0 \Delta x + \sin\theta_0 \Delta y \\ y_2 &= \Delta\theta \\ y_3 &= \sin\theta_0 \Delta x - \cos\theta_0 \Delta y \\ y_4 &= \sin(\varphi_0 - \theta_0)\Delta x + \cos(\varphi_0 - \theta_0)\Delta y - \Delta\theta + \Delta\varphi \end{aligned}$$

The goal of the following computations is to provide a new coordinate system (z_1, z_2, z_3, z_4) at X_0 such that:

- $((f_i)z_k)(X_0) = \delta^i_k$,
- there exists i and j such that $((f_i.f_j)z_3)(X_0) \neq 0$,
- for any i and j, $((f_i.f_j)z_4)(X_0) = 0$, and
- there exists i, j and k such that $((f_h.f_i.f_j)z_4)(X_0) \neq 0$

with $h, i, j \in \{1, 2\}$ and $k \in \{1, 2, 3, 4\}$; $\delta^i_k = 1$ iff $i = k$; (f) designates the differential operator associated to the vector field f; $(f.g)$ is the product of the corresponding differential operators. Such coordinates are called *privileged coordinates.*

One may check that $((f_i)y_k)(X_0) = \delta^i_k$ for $i \in \{1, 2\}$ and $k \in \{1, 2, 3, 4\}$. Moreover $((f_1)^2 y_3)(X_0) = ((f_2)^2 y_3)(X_0) = 0$ and $((f_2.f_1)y_3)(X_0) = 1$. Now, it appears that $((f_1)^2 y_4)(X_0) = \sin\varphi_0 \cos\varphi_0$; then (y_1, y_2, y_3, y_4) is not a privileged coordinate system if $\sin\varphi_0 \cos\varphi_0 \neq 0$.

One gets privileged coordinates by keeping

$$z_1 = y_1, \quad z_2 = y_2, \quad z_3 = y_3$$

and taking

$$z_4 = y_4 - \frac{1}{2}\sin\varphi_0 \cos\varphi_0 y_1^2.$$

In such coordinates, we have

$$f_1 = \begin{pmatrix} \cos z_2 \\ 0 \\ -\sin z_2 \\ F(z_1, z_2, z_3, z_4) \end{pmatrix} \qquad f_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{7}$$

where

$$\begin{aligned} &F(z_1, z_2, z_3, z_4) = -z_1(\cos z_2 \sin 2\varphi_0)/2 + \sin(\varphi_0 + z_2) \\ &- \sin(\varphi_0 - z_1 \sin\varphi_0 + z_1^2(\sin 2\varphi_0)/4 + z_2 + z_3 \cos\varphi_0 + z_4). \end{aligned}$$

The nilpotent approximation is obtained by taking in the Taylor expansions of (7) the terms of homogeneous degree $w_i - 1$ for the i-th coordinate where w_i is the degree of the vector field f_i (i.e., $w_1 = w_2 = 1, w_3 = 2, w_4 = 3$). We get

$$\widehat{f}_1 = \begin{pmatrix} 1 \\ 0 \\ -z_2 \\ \widehat{F}(z_1, z_2, z_3) \end{pmatrix} \qquad \widehat{f}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

where

$$\widehat{F}(z_1, z_2, z_3) = -z_1^2(\sin\varphi_0 \cos 2\varphi_0)/2 - z_1 z_2 \sin^2\varphi_0 - z_3 \cos^2\varphi_0.$$

It is easy to check that this new system is nilpotent of order 3.

4.3 Steering chained form systems

At the same time as Lafferiere and Sussmann work, Murray and Sastry explored in [79, 80] the use of sinusoidal inputs to steer certain nonholonomic systems: the class of systems which can be converted into a chained form. A chained system has the following form:

$$\begin{aligned} \dot{x}_1 &= v \\ \dot{x}_2 &= f_2(x_1)v \\ \dot{x}_3 &= f_3(x_1, x_2)v \\ &\vdots \\ \dot{x}_p &= f_p(x_1, \ldots, x_p)v \end{aligned}$$

with $x_i \in \mathbf{R}^{m_i}$ and $\sum_i m_i = n$.

Because of this special form, there exists simple sinusoidal control that may be used for generating motions affecting the i^{th} set of coordinates while leaving the previous sets of coordinates unchanged. The algorithm then is:

1. Steer x_1 to the desired value using any input and ignoring the evolutions of the x_i's $(1 < i)$,
2. Using sinusoids at integrally related frequencies, iteratively find the inputs steering the x_i's without changing the x_j's, $j < i$.

Extensions [119] by Tilbury and Sastry allow to use sinusoidal control to steer all the coordinates at once for systems with two inputs. They show also how polynomial controls may be used to this end. Moreover Monaco and Normand-Cyrot show that multirate controls (i.e., piece-wise constant controls) provide an exact steering method for chained form systems [76].

Even if a system is not triangular, it may be possible to transform it into a triangular one by feedback transformations (see [80, 81]). Moreover, notice that the nilpotentization techniques introduced in the previous section leads to approximated systems which are in chained form.

Example: Let us consider our canonical example of a mobile robot with two trailers (Figure 4, left). The clever idea which enables the transformation of the system into a chained form was to consider a frame attached to the last trailer rather than attached to the robot [119]. Denoting by θ_1 and θ_2 the angle of the trailers, and by x_2 and y_2 the coordinates of the middle point of the last trailer, the system (6) may be re-written as:

$$\left\{\begin{array}{l} \dot{x} = cos\theta_2 cos(\theta_1 - \theta_2) cos(\theta_0 - \theta_1) u_1 \\ \dot{y} = sin\theta_2 cos(\theta_1 - \theta_2) cos(\theta_0 - \theta_1) u_1 \\ \dot{\theta}_0 = u_2 \\ \dot{\theta}_1 = \sin(\theta_0 - \theta_1) u_1 \\ \dot{\theta}_2 = \sin(\theta_1 - \theta_2) cos(\theta_0 - \theta_1) u_1 \end{array}\right.$$

Let us consider the following change of coordinates:

$$\left\{\begin{array}{l} z_1 = x \\ z_2 = \frac{1}{cos^4\theta_2} \cdot \frac{tan(\theta_0 - \theta_1)}{cos(\theta_1 - \theta_2)} \times (1 + tan^2(\theta_1 - \theta_2)) \\ \quad + \frac{1}{cos^4\theta_2} \times tan(\theta_1 - \theta_2)(3tan(\theta_1 - \theta_2)tan\theta_2 - (1 - tan^2(\theta_1 - \theta_2))) \\ z_3 = \frac{tan(\theta_1 - \theta_2)}{cos^3\theta_2} \\ z_4 = tan\theta_2 \\ z_5 = y \end{array}\right.$$

This transformation is a local diffeomorphism around configurations for which the angle between bodies are not equal to $\frac{\pi}{2}$. In this new coordinates, the kinematic model of the system has the following chained form:

$$\begin{cases} \dot{z}_1 = v_1 \\ \dot{z}_2 = v_2 \\ \dot{z}_3 = z_2.v_1 \\ \dot{z}_4 = z_3.v_1 \\ \dot{z}_5 = z_4.v_1 \end{cases} \tag{8}$$

Notice that Sordalen generalizes this result by providing a conversion of the car with an arbitrary number of trailers into a chained form [109].

Sinusoidal inputs: Let us consider the following inputs [119]:

$$\begin{cases} v_1(t) &= a_0 + a_1 \sin \omega t \\ v_2(t) &= b_0 + b_1 \cos \omega t + b_2 \cos 2\omega t + b_3 \cos 3\omega t \end{cases} \tag{9}$$

Let Z^{start} be a starting configuration. Equations (8) are integrable. Then each $z_i(T)$ can be computed from the five coordinates of Z^{start} and the six parameters $(a_0, a_1, b_0, b_1, b_2, b_3)$. For a given $a_1 \neq 0$ and a given configuration Z^{start}, Tilbury *et al* show that the function computing $Z(T)$ from $(a_0, b_0, b_1, b_2, b_3)$ is a C^1 diffeomorphism at the origin; then the system is invertible and the parameters $(a_0, b_0, b_1, b_2, b_3)$ can be computed from the coordinates of two configurations Z^{start} and Z^{goal}. The system inversion can be done with the help of any symbolic computation software. The corresponding sinusoidal inputs steer the system from Z^{start} to Z^{goal}.

The shape of the path only depends on the parameter a_1. Figure 6 from [99] illustrates this dependence. Moreover the shape of the paths is not invariant by rotation (i.e., it depends on the variables θ^{start} and θ^{goal} and not only on the difference $(\theta^{start} - \theta^{goal})$).

Polynomial inputs: Another steering method is also proposed in [119]. The polynomial inputs:

$$\begin{cases} v_1(t) = 1 \\ v_2(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 \end{cases}$$

steer the system from any configuration Z^{start} to any Z^{goal} verifying $z_1^{goal} \neq z_1^{start}$. In this case T should be equal to $|z_1^{goal} - z_1^{start}|$. As for the case of the sinusoidal inputs, the system can be inverted by symbolic computation. To reach configurations such that $z_1^{goal} = z_1^{start}$ it is sufficient to choose an intermediate configuration respecting the inequality and to apply the steering method twice.

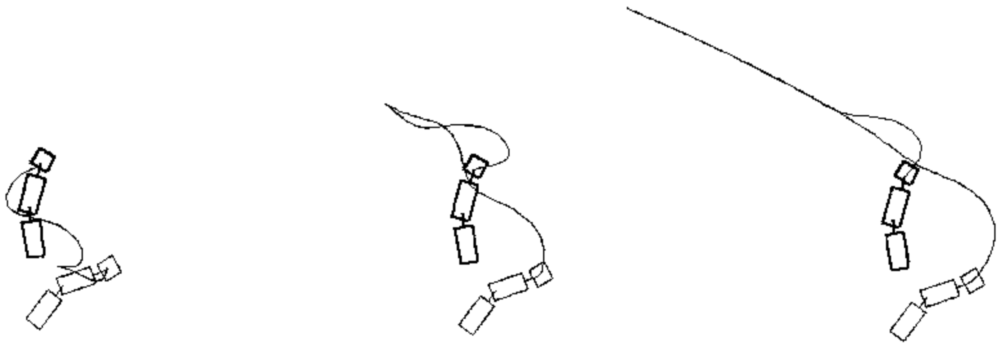


Figure 6: Three paths solving the same problem with three values of a_1: -30, 70, 110

Extensions: The previous steering methods deal with two-input chained form systems. In [22] Bushnell, Tilbury and Sastry extend these results to three-input nonholonomic systems with the fire-truck system as a canonical example[8]. They give sufficient conditions to convert such a system to two-chain, single generator chained forms. Then they show that multirate digital controls, sinusoidal inputs and polynomial inputs may be used as steering methods.

4.4 Steering flat systems

The concept of flatness has been introduced by Fliess, Lévine, Martin and Rouchon [37, 85].

A flat system is a system such that there exists a finite set of variables y_i differentially independent which appear as differential functions of the system variables (state variables and inputs) and of a finite number of their derivatives, each system variable being itself a function of the y_i's and of a finite number of their derivatives. The variables y_i's are called the linearizing outputs of the system.

Example: In [85] Rouchon *et al* show that mobile robots with trailers are flat as soon as the trailers are hitched to the middle point of the wheels of the previous ones. The proof is based on the same idea allowing to transform the

[8] The fire-truck system is a car-like robot (two inputs) with one trailer whose direction of the wheels is controllable (third input).

system into a chained form: it consists in modeling the system by starting from the last trailer.

Let us consider the system (6) (Figure 4, left). Let us denote the coordinates of the robot and the two trailers by (x, y, θ), (x_1, y_1, θ_1) and (x_2, y_2, θ_2) respectively. Remind that the distance between the reference points of the bodies is 1. The holonomic equations allow to compute x, y, x_1 and y_1 from x_2, y_2, θ_1 and θ_2:

$$\begin{aligned} x_1 &= x_2 + \cos\theta_2 \qquad & x &= x_2 + \cos\theta_2 + \cos\theta_1 \\ y_1 &= y_2 + \sin\theta_2 \qquad & y &= y_2 + \sin\theta_2 + \sin\theta_1 \end{aligned}$$

The rolling without slipping conditions lead to three nonholonomic equations $\dot{x_i} \sin\theta_i - \dot{y_i} \cos\theta_i = 0$ allowing to compute θ_2 (resp. θ_1 and θ) from $(\dot{x_2}, \dot{y_2})$ (resp. $(\ddot{x_2}, \ddot{y_2})$ and $(\dddot{x_2}, \dddot{y_2})$). Finally the controls v and ω are given by $v = \frac{\dot{x}}{\cos\theta}$ (or $v = \frac{\dot{y}}{\sin\theta}$) and $\omega = \dot{\theta}$.

Therefore any variable of the system can be computed from x_2 and y_2 and their derivatives. The system is flat with x_2 and y_2 as linearizing outputs.

A steering method: Let us consider a path γ_2 followed by the reference point P_2 of the second trailer (Figure 7). γ_2 is parametrized by arc length s_2. Let us assume that γ_2 is sufficiently smooth, i.e., $\frac{d}{ds_2}P_2$ is defined everywhere and the curvature κ_2 can be differentiated at least once. The point P_1 belongs to the tangent to γ_2 at P_2 and $P_1 = P_2 + \vec{\tau_2}$, with $\vec{\tau_2}$ the unitary tangent vector to γ_2. Differentiating this relation w.r.t. s_2 leads to $\frac{d}{ds_2}P_1 = \vec{\tau_2} + \kappa_2 \vec{\nu_2}$ with $\vec{\nu_2}$ the unitary vector orthogonal to $\vec{\tau_2}$. The angle of the first trailer is then $\theta_1 = \theta_2 + atan(\kappa_2)$. We then deduce the path γ_1 followed by the first trailer. Parametrizing γ_1 with s_1 defined by $ds_1 = (1 + \kappa_2)^{\frac{1}{2}} ds_2$ leads to

$$\kappa_1 = (1 + \kappa_2)^{-\frac{1}{2}} (\kappa_2 + (1 + \kappa_2)^{-\frac{1}{2}}) \frac{d}{ds_2} \kappa_2$$

Applying the same geometric construction from P_1 we can compute the path γ followed by the robot when the second trailer follows γ_2. The only required condition is the existence of $\frac{d^2}{ds_2^2}\kappa_2$; moreover the relative angles φ_1 and φ_2 should belong to $]-\frac{\pi}{2}, \frac{\pi}{2}[$ (see [37] for details).

Two configurations X^{start} and X^{goal} being given, one computes geometrically the values of κ^{start}, κ_1^{start}, κ_2^{start}, κ^{goal}, κ_1^{goal} and κ_2^{goal}; each of them being a function of κ_2 and its derivative, it is straightforward to compute γ_2 satisfying such initial and final conditions (e.g., by using polynomial curves).

Remark: Because the curvature of γ_2 should be defined everywhere, the method can not provide any cusp point; nevertheless such points are required in some situations like the parking task; in that case, Rouchon *et al* enter the cusp point by hand [85]. We will see below how to overcome this difficulty.

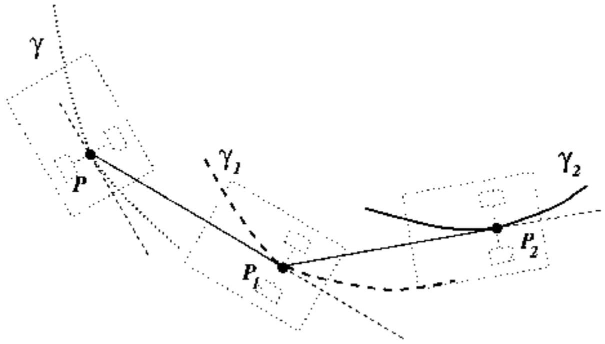


Figure 7: Geometric construction of P_1 (resp. P) path from P_2 (resp. P_1) path.

Flatness conditions: In cite [86], Rouchon gives conditions to check whether a system is flat. Among them there is a necessary and sufficient condition for two-input driftless systems: it regards the rank of the various vector space Δ^k iteratively defined by $\Delta_0 = span\{f_1, f_2\}$, $\Delta_1 = span\{f_1, f_2, [f_1, f_2]\}$ and $\Delta_{i+1} = \Delta_0 + [\Delta_i, \Delta_i]$ with $[\Delta_i, \Delta_i] = span\{[f, g]\ ,\ f \in \Delta_i,\ g \in \Delta_i\}$. A system with two inputs is flat iff $rank(\Delta_i) = 2 + i$.

Let us apply this condition to the mobile robot system with two trailers. According to the computations presented in Section 2.3:

- for the case shown in Figure 4 (left), we get:

  $$rank(\Delta_0) = 2,\ \ rank(\Delta_1) = 3,\ \ rank(\Delta_2) = 4 \text{ and } rank(\Delta_3) = 5$$

  the system is flat.

- for the case shown in Figure 4 (right), we get:

  $$rank(\Delta_0) = 2,\ \ rank(\Delta_1) = 3 \text{ and } rank(\Delta_2) = 5$$

  the system is not flat.

- for the same case shown in Figure 4 (right) but with only one trailer, one can check that:

  $$rank(\Delta_0) = 2,\ \ rank(\Delta_1) = 3 \text{ and } rank(\Delta_2) = 4$$

  the system is flat.

We have seen that the linearizing outputs in the first case are the coordinates of the reference point of the second trailer. In the last case, the linearizing outputs are more difficult to translate into geometric terms (see [85]). Notice that there is no general method to compute the linearizing outputs when the system is flat.

4.5 Steering with optimal control

Optimal length paths have been at the origin of the very first nonholonomic motion planners for car-like mobile robots (see for instance [67, 60] and below). Nevertheless, today the only existing results allowing to compute optimal paths for nonholonomic systems have been obtained for the car-like systems (see Annex C). For general systems, the only possibility is to call on numerical methods.

We sketch here the method developed by Fernandez, Li and Gurvitz in [35].

Let us consider a dynamical system: $\dot{X} = B(X)u$ together with a cost function $J = \int_0^T < u(\tau), u(\tau) > d\tau$. Both starting and goal configurations being given, the optimization problem is to find the control law (if any) that steers the system from the starting configuration to the goal in time T by minimizing the cost function J. The path corresponding to an optimal control law is said to be an optimal path.

Let us consider a continuous and piecewise C^1 control law u defined over $[0, T]$. We denote by $\tilde{u}$ the periodic extension of u over $\mathbf{R}$. We may write $\tilde{u}$ in terms of a Fourier basis:

$$\tilde{u} = \sum_{k=0}^{\infty}(\alpha_k e^{i\frac{2k\pi t}{T}} + \beta_k e^{-i\frac{2k\pi t}{T}})$$

We then approximate $\tilde{u}$ by truncating its expansion up to some rank N. The new control law $\hat{u}$ is then defined by N real numbers[9] $\hat{u} = \sum_{k=1}^{N} \alpha_k e_k$, $e_k \in \{e^{i\frac{2p\pi}{T}}, p \in \mathbf{Z}\}$. The choice of the reals α_k being given, the point $X(T)$ reached after a time T with the control law $\hat{u}$ appears as a function $f(\alpha)$ from $\mathbf{R}^N$ to $\mathbf{R}^n$.

Now, we get a new cost function

$$\hat{J}(\alpha) = \sum_{k=1}^{N} |\alpha_k|^2 + \gamma \|X(T) - X_{goal}\|^2.$$

The new optimization problem becomes: for a fixed time T, an initial point X_{init} and a final point X_{goal} , find $\alpha \in \mathbf{R}^N$ such as

$$\hat{J}(\alpha) = \sum_{k=1}^{N} |\alpha_k|^2 + \gamma \|f(\alpha) - X_{goal}\|^2$$

is minimum.

One proves (*e.g.*, see [35]) that the solutions of the new finite-dimensional problem converge to the solutions of the original problem as N and γ go to infinity.

[9] This approximation restricts the family of the admissible control laws.

Because we do not know f and $\delta f/\delta\alpha$ explicitly we use numerical methods (numerical integration of the differential equations and numerical optimization like Newton's algorithm) to compute a solution of the problem. Such a solution is said to be a near-optimal solution of the original problem.

Figure 8 from [99] shows three examples of near-optimal paths computed from this method for a mobile robot with two trailers [68].

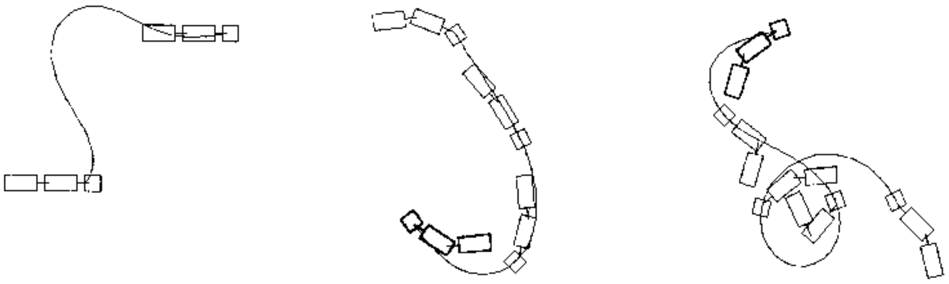


Figure 8: Three examples of near-optimal paths.

5 Nonholonomic path planning for small-time controllable systems

Consider the following steering method for a two-driving wheel mobile robot. To go from the origin $(0,0,0)$ to some configuration (x,y,θ) the robot first executes a pure rotation to the configuration $(0,0,atan\frac{y}{x})$, then it moves along a straight line segment to $(x,y,atan\frac{y}{x})$, and a final rotation steers it to the goal. This simple method accounts for local controllability: any point in any neighborhood of the origin can be reached by this sequence of three elementary paths (when $x=0$, replace $atan\frac{y}{x}$ by $\pm\frac{\pi}{2}$). Nevertheless such a method does not account for small-time controllability. If the space is very constraint it does not hold. Think to the parking task (Figure 19): the allowed mobile robot orientations θ vary in some interval $]-\eta,\eta[$. To go from $(0,0,0)$ to $(0,\epsilon,0)$ the steering method violates necessarily this constraint.

Therefore, obstacle avoidance requires steering methods accounting for small-time controllability. Such a requirement can be translated into geometric terms.

5.1 Toward steering methods accounting for small-time controllability

Let $\mathrm{d}_{\mathcal{CS}}$ be the following distance over the configuration space $\mathcal{CS}$:

$$\mathrm{d}_{\mathcal{CS}}(X^1, X^2) = \sum_{i=1}^{n} |x_i^1 - x_i^2|$$

The set of configurations X^2 such that $\mathrm{d}_{\mathcal{CS}}(X^1, X^2) < \epsilon$ is denoted by $B(X^1, \epsilon)$; this is the ball centered at X^1 with radius ϵ.

Let $\mathcal{P}$ be the set of feasible paths defined over an interval of the type $[0, T]$. A steering function is a mapping from $\mathcal{CS} \times \mathcal{CS}$ into $\mathcal{P}$:

$$(X^1, X^2) \rightarrow \mathsf{Steer}(X^1, X^2)$$

where $\mathsf{Steer}(X^1, X^2)$ is defined over the interval $[0, T]$, such that $\mathsf{Steer}(X^1, X^2)(0) = X^1$, $\mathsf{Steer}(X^1, X^2)(T) = X^2$.

Definition 2 Steer *verifies the weak topological property iff:*

$$\forall \epsilon > 0, \forall X^1 \in \mathcal{CS}, \exists \eta > 0, \forall X^2 \in \mathcal{CS}, \tag{10}$$
$$d_{\mathcal{CS}}(X^1, X^2) < \eta \;\Rightarrow\; \forall t \in [0, T],\;\; d_{\mathcal{CS}}(\mathsf{Steer}(X^1, X^2)(t), X^1) < \epsilon$$

By using a steering method that verifies the weak topological property, it is possible to approximate any collision-free path γ_{free}. Nevertheless, this property is not sufficient from a *computational* point of view. Indeed, it is *local*: the real number η depends on X^1. Situations as shown in Figure 9 may appear: let us consider a sequence of configurations X^i converging to the critical point X^c, and such that $\lim_{X^i \to X^c} \eta(X^i) = 0$; to be collision-free any admissible path should necessary goes through the configuration X^c. The computation of X^c may set numerical problems. To overcome this difficulty, we introduce a stronger property for the steering methods.

Definition 3 Steer *verifies the topological property iff:*

$$\forall \epsilon > 0, \exists \eta > 0, \forall (X^1, X^2) \in (\mathcal{CS})^2, \tag{11}$$
$$d_{\mathcal{CS}}(X^1, X^2) < \eta \;\Rightarrow\; \forall t \in [0, T],\;\; d_{\mathcal{CS}}(\mathsf{Steer}(X^1, X^2)(t), X^1) < \epsilon$$

In this definition η does not depend on any configuration (Figure 10). This is a global property that not only accounts for small-time controllability but also holds uniformly everywhere.

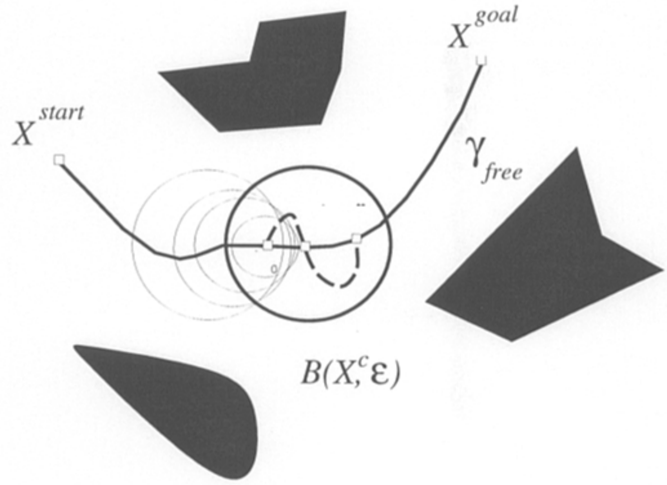


Figure 9: Weak topological property

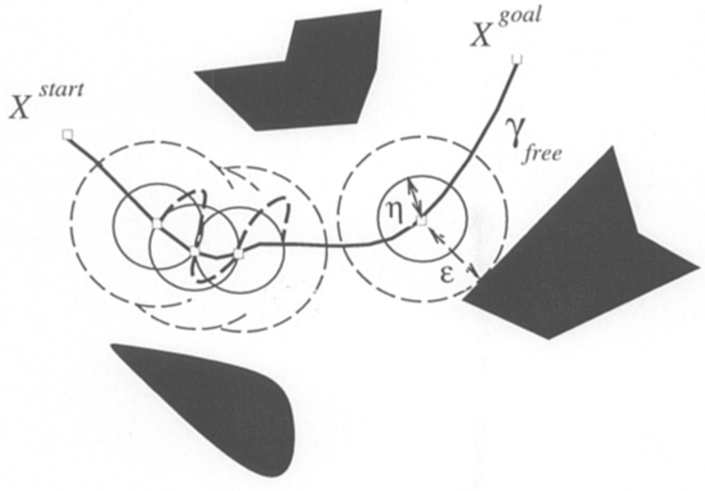


Figure 10: Topological property

Remark 1: Proving that a given steering method verifies the topological property is not an easy task. The following sufficient condition appears in [99, 102]. Let us equip $\mathcal{P}$ with a metric $d_{\mathcal{P}}$ between paths: Γ_1 and Γ_2 being two paths on $[0,1]$, we define $d_{\mathcal{P}}(\Gamma_1, \Gamma_2) = \max_{t\in[0,1]} \mathrm{d}_{\mathcal{CS}}(\Gamma_1(t), \Gamma_2(t))$.

Let us consider a steering method Steer continuous w.r.t. to the topology induced by $d_{\mathcal{P}}$. Steer is uniformly continuous on any compact set $\mathcal{K}$ included in $\mathcal{CS}^2$, i.e.,

$$\forall\epsilon > 0\ ,\ \exists\eta > 0,\ \ \forall\ \{(X^1, X^2), (Y^1, Y^2)\} \in \mathcal{K}$$

$$d_{CS}((X^1, Y^1) < \eta \text{ and } d_{CS}((X^2, Y^2) < \eta \implies d_{\mathcal{P}}(\mathsf{Steer}(X^1, X^2), \mathsf{Steer}(Y^1, Y^2)) < \epsilon$$

Choosing $X^2 = Y^1 = Y^2$ we deduce:

$$\forall\epsilon > 0\ ,\ \exists\eta > 0,\ \ \forall(X^1, X^2) \in \mathcal{CS}^2$$

$$d_{CS}(X^1, X^2) < \eta \Longrightarrow d_{\mathcal{P}}(\mathsf{Steer}(X^1, X^1), \mathsf{Steer}(X^1, X^2)) < \epsilon$$

Now, let us assume that $\mathsf{Steer}(X, X) = \{X\}$ at any point X. Then:

$$\forall\epsilon > 0\ ,\ \exists\eta > 0\ \ \forall(X^1, X^2) \in \mathcal{CS}^2$$

$$d_{CS}(X^1, X^2) < \eta \implies \forall t \in [0,1],\ \ d_{CS}(X^1, \mathsf{Steer}(X^1, X^2)(t)) < \epsilon$$

Therefore a sufficient condition for a steering method Steer to verify the topological property is that (1) Steer is continuous w.r.t. the topology associated with $d_{\mathcal{P}}$ and (2) the path $\mathsf{Steer}(X, X)$ is reduced to the point X. Notice that the second condition is obviously a necessary condition.

Remark 2: The first general result taking into account the necessary uniform convergence of steering methods is due to Sussmann and Liu [115]: the authors propose an algorithm providing a sequence of feasible paths that *uniformly* converge to any given path. This guarantees that one can choose a feasible path *arbitrarily close* to a given collision-free path. The method uses high frequency sinusoidal inputs. Though this approach is general, it is quite hard to implement in practice. In [120], Tilbury *et al* exploit the idea for a mobile robot with two trailers. Nevertheless, experimental results show that the approach cannot be applied in practice, mainly because the paths are highly oscillatory. Therefore this method has never been connected to a geometric planner in order to get a global planner which would take into account both environmental and kinematic constraints.

5.2 Steering methods and topological property

In this section we review the steering methods of Section 4 with respect to the topological property.

5.2.1 Optimal paths

Let us denote by Steer_{opt} the steering method using optimal control. Steer_{opt} naturally verifies the weak topological property. Indeed the cost of the optimal paths induce a special metric in the configuration space; such a metric is said to be a nonholonomic [126], or singular [18], or Carnot-Caratheory [75], or sub-Riemannian [111] metric. By definition any optimal path with cost r does not escape the nonholonomic ball centered at the starting point with radius r. A general result states that the nonholonomic metrics induce the same topology as the "natural" metrics d_{CS}. This means that for any nonholonomic ball $B_{nh}(X,r)$ with radius r, there are two real numbers ϵ and η strictly positive such that $B(X,\eta) \subset B_{nh}(X,r) \subset B(X,\epsilon)$.

The nonholonomic distance being continuous, to get the topological property, it suffices to restrict the application of Steer_{opt} to a compact domain of the configuration space[10].

There is no general result that gives the exact shape of the nonholonomic balls; nevertheless the approximated shape of these balls is well understood (e.g., see [12]).

The metric induced by the length of the shortest paths for Reeds&Shepp's car is close to a nonholonomic metric; car-like robots are the only known cases where it is possible to compute the exact shape of the balls (see Figure 3).

5.2.2 Sinusoidal inputs and chained form systems

Let us consider the two-input chained form system (8) together with the sinusoidal inputs (9) presented in Section 4.3. We have seen that the shape of the paths depends on a_1 (Figure 6). The only constraint on the choice of a_1 is that it should be different from zero.

The steering method using such inputs is denoted by $\mathsf{Steer}^{a_1}_{sin}$. For a fixed value of a_1, $\mathsf{Steer}^{a_1}_{sin}$ does not verify the topological property. Indeed, for any configuration Z, the path $\mathsf{Steer}^{a_1}_{sin}(Z,Z)$ is not reduced to a point[11].

Therefore, the only way to build a steering method based on sinusoidal inputs and verifying the topological property is not to keep a_1 constant. One has to prove the existence of a continuous expression of $a_1(Z^1,Z^2)$ such that:

$$\begin{aligned}
\lim_{Z^2 \to Z^1} a_1(Z^1,Z^2) &= 0 \\
\lim_{Z^2 \to Z^1} a_0(Z^1,Z^2,a_1(Z^1,Z^2)) &= 0 \\
\lim_{Z^2 \to Z^1} b_i(Z^1,Z^2,a_1(Z^1,Z^2)) &= 0
\end{aligned}$$

[10] Notice that the steering method Steer_{opt} is not necessarily continuous w.r.t. the topology induced by $d_{\mathcal{P}}$.

[11] The first coordinate of points lying on the path is $z_1(t) = z_1 + \frac{a_1}{\omega}(1 - \cos\omega t)$.

The proof appears in [101]. It first states that, for a fixed value of a_1, $\mathsf{Steer}_{sin}^{a_1}$ is continuous w.r.t. to the topology induced by $d_{\mathcal{P}}$. This implies that when the final configuration Z^2 tends to the initial configuration Z^1, the path $\mathsf{Steer}_{sin}^{a_1}(Z^1, Z^2)$ tends to the path $\mathsf{Steer}_{sin}^{a_1}(Z^1, Z^1)$. Moreover, for any Z, $\mathsf{Steer}_{sin}^{a_1}(Z, Z)$ tends to $\{Z\}$ when a_1 tends to zero. Combining these two statements and restricting the application of $\mathsf{Steer}_{sin}^{a_1}$ to a compact domain $\mathcal{K}$ of $\mathcal{CS}^2$, one may conclude that:

$$\forall \epsilon > 0\ ,\ \exists A_1 > 0\ \ \forall a_1 < A_1,\ \exists \eta(a_1),\ \ \forall (Z^1, Z^2) \in \mathcal{K},$$

$$d_{CS}(Z^1, Z^2) < \eta(a_1),\ \implies\ \forall t \in [0,1],\ \ d_{CS}(Z^1, \mathsf{Steer}_{sin}^{a_1}(Z^1, Z^2)(t)) < \epsilon$$

Then, by tuning a_1, it is *a priori* possible to design a steering method Steer_{sin} based on sinusoidal inputs and verifying the topological property. It remains to define a constructive way to tune the parameter a_1. The problem is not easy. Indeed the general expression of parameters a_0 and b_i are unknown. Then we do not dispose of a unique expression of Steer_{sin}. Nevertheless, it is possible to "simulate" a steering method verifying the topological property, by switching between different $\mathsf{Steer}_{sin}^{a_1}$ according to the distance between the start and goal configurations. The principle of the construction presented in Annex D consists in introducing the possibility to iteratively compute subgoals and then a sequence of subpaths that reaches the final goal without escaping a bounded domain.

5.2.3 A flatness based steering method for mobile robots with trailers

We have seen in Section 4.4 that a mobile robot with two trailers (with centered hooking up system) is flat with the coordinates (x_2, y_2) of the second trailer reference point P_2 as linearizing outputs. Planning an admissible path for the system then consists in finding a sufficiently smooth planar curve $\gamma_2(s)$ for P_2. All the coordinates $(x, y, \theta, \varphi_1, \varphi_2)$ of a configuration can be geometrically deduced from $(x_2, y_2, \theta_2, \kappa_2, \frac{d}{ds_2}\kappa_2)$. Nevertheless this steering method cannot verify the topological properties. Indeed, due to the conditions on γ_2 (absence of cusp points), going from a configuration $(x_2, y_2, \theta_2, \ .\ ,\ .\ )$ to some configuration $(x_2, y_2 + \epsilon, \theta_2, \ .\ ,\ .\ )$ should necessarily contain a configuration $(\ .\ ,\ .\ , \theta_2 \pm \frac{\pi}{2}, \ .\ ,\ .\ )$.

[102] takes advantage of the flatness property of a mobile robot with one trailer to design a steering method verifying the topological property. [56] generalizes the method to a system with n trailers. Let us sketch the method for a mobile robot with two trailers.

Let us consider a configuration $X = (x, y, \theta, \varphi_1, \varphi_2)$ of the system. If Γ is an admissible path in the configuration space, we will denote by γ_2 the curve in $\mathbf{R}^2$ followed by P_2. Among all the admissible paths containing a configuration X, there exists exactly one path Γ such that $\frac{d}{ds_2}\kappa_2$ remains constant everywhere: the corresponding curve γ_2 is a clothoid[12].

Let X^{start} and X^{goal} be the initial and goal configurations respectively. Let $\gamma_{2,start}$ and $\gamma_{2,goal}$ be the associated clothoids defined on $[0,1]$ and such that $\Gamma_{start}(0) = X^{start}$ and $\Gamma_{goal}(1) = X^{goal}$. Then any combination $\gamma(t) = \alpha(t)\gamma_{2,start}(t) + (1-\alpha(t))\gamma_{2,goal}(t)$ is a C^3 path; it then corresponds to an admissible path for the whole system. To make this path starting at X^{start} and ending at X^{goal}, α should verify: $\alpha(0) = \dot{\alpha}(0) = \ddot{\alpha}(0) = \dddot{\alpha}\ (0) = \dot{\alpha}(1) = \ddot{\alpha}(1) = \dddot{\alpha}\ (1) = 0$ and $\alpha(1) = 1$ (indeed, the three first derivatives of γ should be the same as those of $\gamma_{2,start}$ at 0 and the same as those of $\gamma_{2,goal}$ at 1).

At this level we get a steering method (denoted by Steer^*_{flat}) that allows the mobile robot with its two trailers to reach any configuration from any other one. Nevertheless, this method does not verify the required topological property: indeed it cannot generate cusps and the steering method we want to provide should be able to do it when it is necessary.

Let X^{start} be an initial configuration and $\gamma_{2,start}$, the corresponding clothoid in $\mathbf{R}^2$. In [56] we prove that, for any $\epsilon > 0$, if we choose a configuration X within a "cone" $C_{start,\epsilon}$ centered around Γ_{start} with vertex X^{start} (see Figure 11), then the path $\mathsf{Steer}^*_{flat}(X^{start}, q)$ does not escape the ball $B(X^{start}, \epsilon)$.

Moreover, if X^{start} moves within a small open set, the clothoid $\gamma_{2,start}$ is submitted to a continuous deformation: for instance a change on the coordinates (x_2, y_2) (respectively θ_2) of X^{start} corresponds to a translation (respectively rotation) of the clothoid $\gamma_{2,start}$. Then for a small deformation, the corresponding path in the configuration space necessarily intersects $C_{start,\epsilon}$.

Let us now consider a configuration X^{goal} sufficiently close to X^{start}. The local planner Steer_{flat} then works as follows:

- If X^{goal} belongs to $C_{start,\epsilon}$, $\mathsf{Steer}_{flat}(X^{start}, X^{goal}) = \mathsf{Steer}^*_{flat}(X^{start}, X^{goal})$
- otherwise, we choose a point X^{cusp} on $\gamma_{2,goal}$ within $C_{start,\epsilon}$ and $\mathsf{Steer}_{flat}(X^{start}, X^{goal})$ is constituted by $\mathsf{Steer}^*_{flat}(X^{start}, X^{cusp})$ followed by the arc of the clothoid $\gamma_{2,goal}$ between X^{cusp} and X^{goal}.

With this construction $\mathsf{Steer}_{flat}(X^{start}, X^{goal})$ is guaranteed to remain within the ball $B(X^{start}, \epsilon)$ (Figure 11).

Figure 12 shows an example of the path generated by Steer_{flat}.

Figures 13 shows a comparison between paths computed by Steer_{sin} and Steer_{flat} for a mobile robot with one trailer (the curve is the path followed by the robot).

[12] If we consider only one trailer, we impose κ_1 to remain constant; in this case the trailer follows a circle with radius $\ell.\mathrm{cotan}\varphi_1$

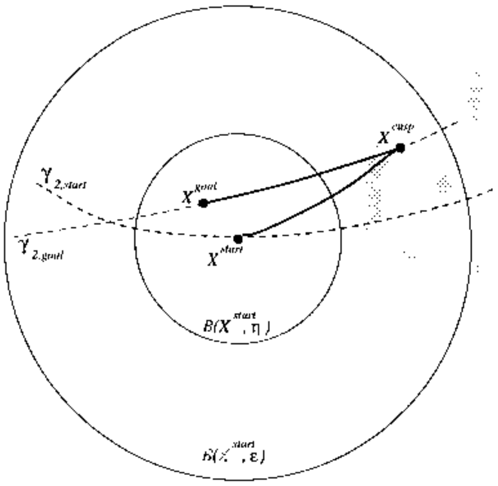


Figure 11: Topological property of Steer_{flat}.

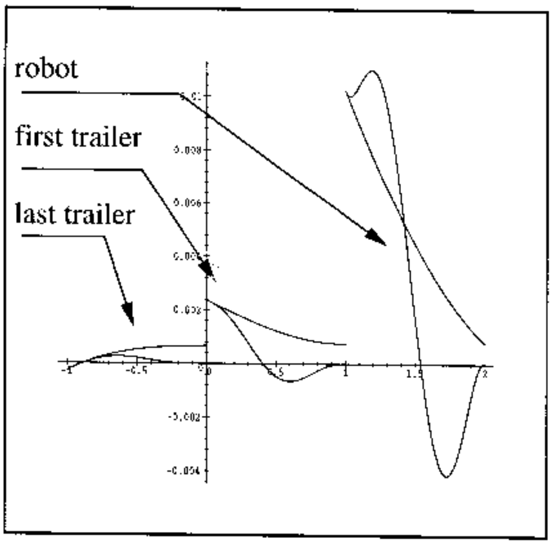


Figure 12: An admissible path for a mobile robot with 2 trailers

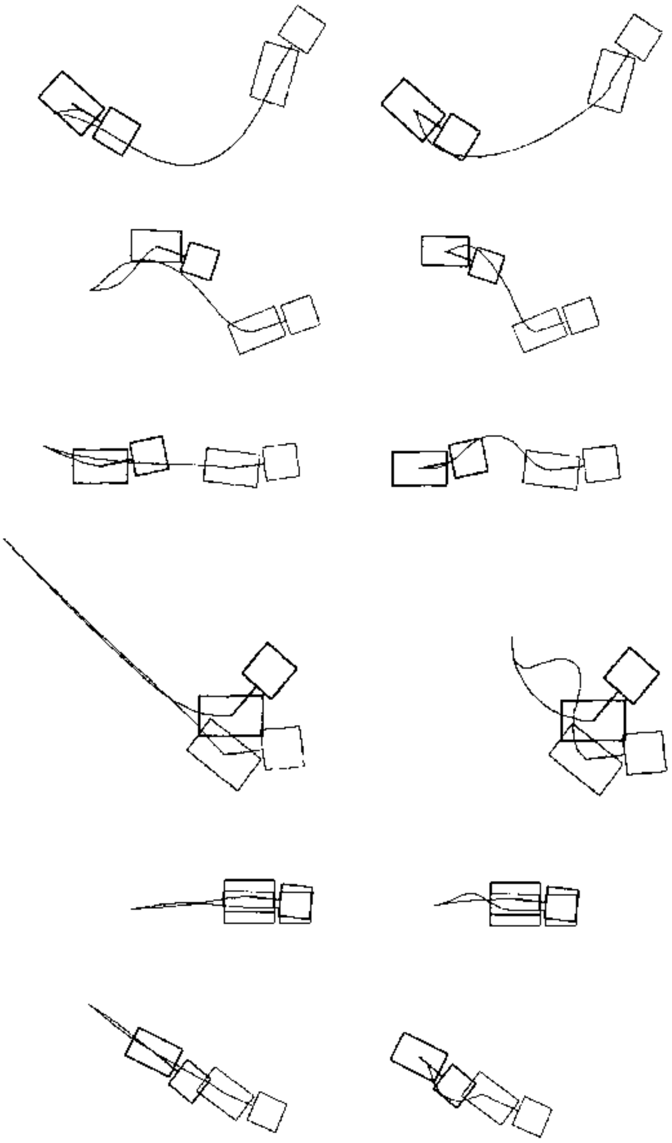


Figure 13: Comparison between two steering methods on System A: sinusoidal inputs for the equivalent chained system (left) and flatness-based steering method (right). Initial and final configurations are the same in each of the six cases. Note that Steer_{flat} generates more "natural" paths requiring fewer cusp and space than the sinusoidal inputs.

5.3 Approximating holonomic paths: a two step approach

5.3.1 Principle

Everything being in place, we may now define a first nonholonomic path planning scheme for small-time controllable systems. It consists in approximating a collision-free (holonomic) path by a sequence of collision-free admissible ones. Applying this scheme requires three main components:

- A geometric path planner that computes collision-free paths without taking into account the kinematic constraints.
- A steering method verifying the topological property.
- A geometric routine checking whether a given path is collision-free or not.

The algorithm itself is then very simple:

1. Step 1: Plan a collision-free path with the geometric path planner. If no such path exists, the algorithm stops: there is no solution.
2. Step 2: Perform subdivisions on the path until all endpoints can be linked to their neighbors by an admissible collision-free path.

Convergence and completeness: By Theorem 2, the convergence of Step 2 is guaranteed as soon as the steering method verifies the topological property. Then the completeness of the algorithm only depends on the completeness of the geometric planner that computes a first collision-free path[13].

Geometric planner: There are no general and practical algorithm solving the classical "piano mover" problem with completeness property[14]. Numerous techniques are available to address dedicated problems [59]. Moreover new general principles appeared in the past few years. Among them one should notice the "distributed representation approach" [5] that leads to resolution-complete algorithms (such algorithms are guaranteed to find a solution when a solution exists at a given resolution when modeling the search space by a grid). Another notion is related to the behavior of probabilistic algorithms: an algorithm is said to be probabilistically complete if it includes random choices and if it is guaranteed to find a solution in finite

[13] An algorithm is complete if it is guaranteed to report a negative answer when a solution does not exist and to compute a solution otherwise.

[14] General algorithms have been provided within the framework of real algebraic geometry [97, 24]; nevertheless the performance of the existing algebraic computing software and the intrinsic computational complexity of the general problem do not allow effective implementations of these algorithms.

(possibly unbounded) time when a solution exists; such algorithms cannot terminate with a negative answer on the existence of a solution. Nevertheless resolution and probabilistically complete algorithms are well understood [8] and they lead today to fast and practical motion geometric planners even for highly dimensionated systems.

Smoothing step: Step 2 provides a sequence of elementary admissible paths computed by the chosen steering method. The length of the sequence mainly depends on the clearance of the first collision-free path: the closer to the obstacles the path is, the more it should be subdivided. The sequence may be shorten in a third step by applying the steering method to link randomly chosen pairs of points lying on the first solution path. Unfortunately there is today no result insuring the convergence of this third step to any "optimal" sequence.

Several nonholonomic path planners have been designed in this way. Here is a review of the main ones.

5.3.2 Application to mobile robots (without trailer)

The seminal ideas of the algorithm above have been introduced in [67]. This reference deals with car-like robots. When the robot is a polygon, the geometric planner is derived from the algorithm based on an analytical representation of the configuration space of a polygon moving amidst polygonal obstacles [4]. When the robot is a disk, the geometric planner works from the Voronoi diagram of the environment. In both cases, the steering method is Steer_{opt}: it consists in computing the shortest length admissible paths for a car-like robot as characterized in [87]. Due to the completeness of the geometric planners the proposed algorithms are complete. Nevertheless they are delicate to implement and fragile in practice (indeed the basic geometric routines are sensitive to numerical computations; a robust implementation could be done by using software computing with rational numbers).

Another version of this algorithm appears in [60] where the geometric planner has been replaced by a distributed representation approach; the search consists in exploring the discretized configuration space with an A^* algorithm heuristically guided by a potential function. It is then resolution-complete, less fragile than the original version, and sufficiently efficient to be integrated on real robots. Figure 14 shows an example of a solution from a software developed for the mobile robots Hilare at LAAS.

A clever idea appears in [74]. It tends to minimize the length of the shortest path sequence approximating the geometric path. It consists in computing a skeleton gathering the set of points with maximum clearance with respect to the obstacles. The key point is that the clearance is measured with the metric induced by the length of the shortest admissible paths (the so-called Reeds&Shepp metric). Then the geometric planner works by

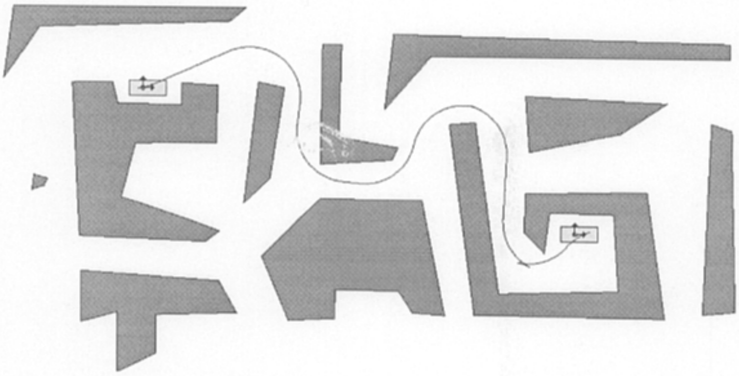


Figure 14: A planned path for a car-like robot: the workspace is modeled by a grid of 250 × 150 pixels; the total running of the algorithm is 2 seconds on an Indy Silicon Graphics.

retracting the initial and goal configurations on the skeleton and by exploring it. Even if one cannot conclude to any optimality of the solution, the sequence of shortest paths provided the approximation step is shorter than a sequence approximating a path that would lie closer to the obstacles. A critical point of the approach is the computation of the metric. The distance between two configurations being invariant by translation and rotation, the authors use a lookup table storing all the distance value over a discretized compact region around the origin. This table is computed off-line once and may be used to compute the skeleton of various environments.

Recent results derived from the synthesis of the shortest paths for a car-like robot (see Annex C) provide an analytical way to compute the shortest path distance to a polygonal obstacle for a point car-like robot [124]. This means that all the distance computations in the Reeds&Shepp metric can be done on-line. This property has been exploited to include dynamic obstacle avoidance when the robot executes its trajectory. Figure 15 from [53] shows an example of on-line updating of an admissible path when an unexpected obstacle (the black box) occurs during the execution of the motion. The various balls covering the path in the figure are the projection onto $\mathbf{R}^2$ of the maximal collision-free Reeds&Shepp balls covering the path in the configuration space. Up to now, the distance function are known for a point robot; its extension to a polygonal robot has to be done.

5.3.3 The case of mobile robots with trailers

The case of mobile robots with n trailers has been solved by using RPP as geometric planner and the three steering methods Steer_{opt}, Steer_{sin} (for $n = 1$ and $n = 2$) and Steer_{flat} (for $n = 1$) [99].

To compute a collision-free path we use the algorithm RPP, the random

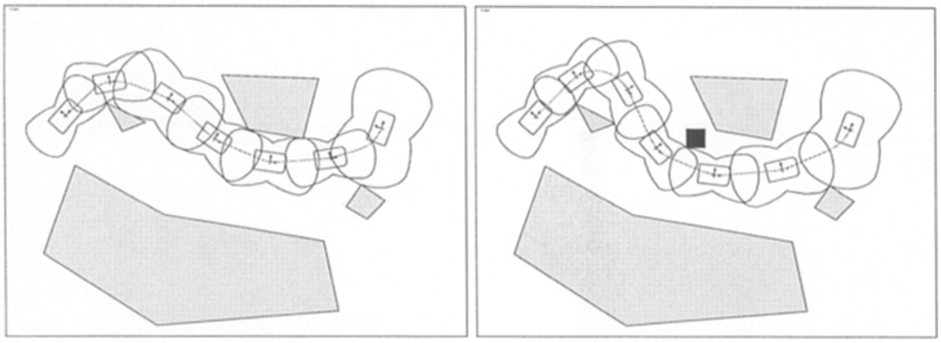


Figure 15: A planned path is updated in real-time when an unexpected moving obstacle occurs.

path planner presented in [5]. We consider $n+1$ control points: two are located on the robot and one is located on each trailer. The start configuration and the goal being given, a potential field is computed for each control point in the workspace[15]; the $n+1$ potential fields are then combined to create a potential field in the configuration space; the search consists in following the gradient of the potential; when it stops at some local minimum, the algorithm generates a random path and follows again the gradient until the goal is reached. The algorithm is probabilistically complete.

From the various experiments reported in [99], it appears that the the algorithm based on Steer_{sin} is much faster than the algorithm based on Steer_{opt} (in terms of computation time) for a mobile robot with one or two trailers. For a mobile robot with one trailer the computation time are roughly the same when using Steer_{flat} and Steer_{sin}; nevertheless the smoothness of the final path is better with Steer_{flat} than with Steer_{sin}.

In the examples on Figures 16 and 17 the workspace is modeled by a grid of 600 × 470 pixels.

5.4 Probabilistic approaches

Recent results have been provided by applying a new general paradigm in motion planning. This is a probabilistic approach consisting in two phases (see [118] for an overview):

- In a first learning phase an incremental roadmap is built by randomly choosing collision-free configurations and by linking them with admissible paths. Admissible paths are computed with a (not necessarily complete) local path planner.

- In the query phase, paths are to be found between some given start and goal configurations. The local path planner is used to connect the

[15] The potential field are computed from a bitmap representation of the workspace.

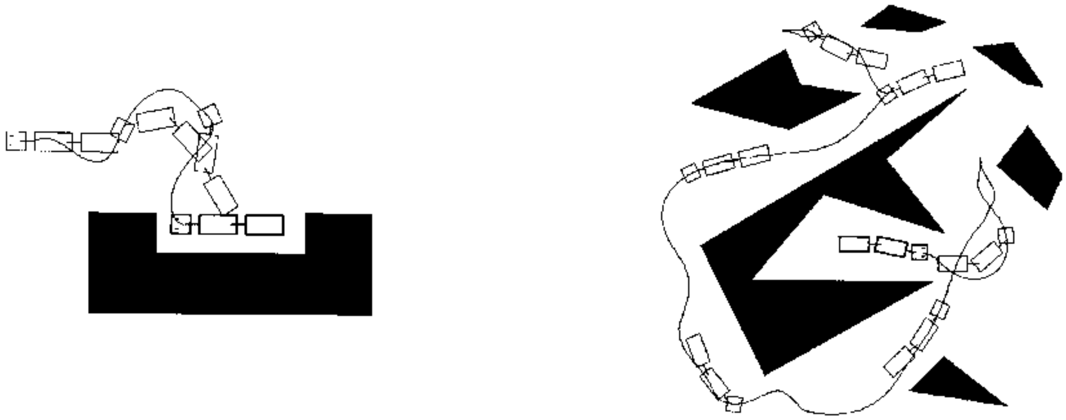


Figure 16: Solutions using Steer$_{sin}$: the total computation time is 30 seconds (left) and 114 seconds (right) on a Sun-Sparc-20.

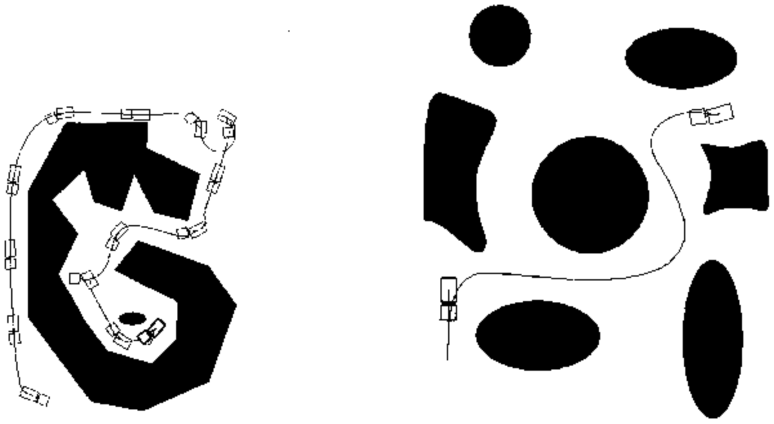


Figure 17: Solutions using Steer$_{flat}$: the total computation time is 21 seconds (left) and 6 seconds (right) on a Sun-Sparc-20.

configurations to some nodes of the roadmap. If this succeeds, a graph search is performed.

As for the approach using a holonomic path approximation, the algorithm includes a last step consisting in smoothing the computed solution.

Such a scheme applies for nonholonomic systems as soon as the local path planner is a steering method verifying the topological property. The algorithm is probabilistically complete. It has been applied to mobile robots with trailers on the basis of Steer_{sin} [98]. An analysis of the approach together with practical results are overviewed in Svestka–Overmars' chapter.

5.5 An approach using optimization techniques

At the same time, a slightly different approach has been proposed by Bessière *et al* [14]. Its principle consists in exploring the free space from the initial configuration along admissible paths by spreading landmarks, each being as far as possible from one another. In parallel, a local path planner checks if the target may be reached from each new landmark. Both phases are solved by using optimisation techniques (e.g., genetic algorithms). This general paradigm has been applied to nonholonomic mobile robots in [3] by using the Steer_{sin} as local path planner. Because Steer_{sin} verifies the topological property the algorithm may be proved to be complete as soon as the convergence the optimization routines is guaranteed.

5.6 A multi-level approach

It remains that the computational cost of the nonholonomic path planners increases with the dimension of the systems. Facing the intrinsic complexity of the problem for practical applications requires a good understanding of the kinematic structures of the systems as well as a good experience in evaluating the performance of a given planning scheme. [98] presents a multi-level nonholonomic path planner.

Let us illustrate the idea from a car-like robot pulling two trailers: from the collision avoidance point of view the system is of dimension five (three parameters for the car and one parameter for each trailer); from the control point of view the direction of the front wheels of the car is taken into account: the system is then six-dimensionated.

The underlying idea consists in introducing the nonholonomic constraints of the bodies *iteratively*. In a first step one plans a "semi-holonomic" path feasible by the car, but not necessarily by the trailers (i.e. at this step the trailers are assumed to be holonomic). Then the nonholonomic constraint due to the first trailer is introduced: this step consists in searching a path feasible by both the car and the first trailer. Finally, all the kinematic constraints are taken into account.

Each step should benefit from the path computed by the previous one, via a specific nonholonomic motion planner. In [98], the first semi-holonomic path is computed with a probabilistic approach that considers only the kinematic constraints of the car. Then a probabilistic search using Steer_{sin} is applied within a tube surrounding the path; it provides a second semi-holonomic path that takes into account the first three kinematic constraints. Finally the second path is approximated via Steer_{sin} accounting for all the constraints. The global algorithm is then based on a combination of the holonomic path approximation scheme and the probabilistic one.

Three examples of solutions provided by the algorithm appear in Figure 18: the left column shows the first "semi-holonomic" paths (the two trailers "slide"); the right column shows the corresponding final paths. The total time to compute the solutions ranges from less than one minute for the first example to around three minutes for the third one, on a 136 MIPS workstation.

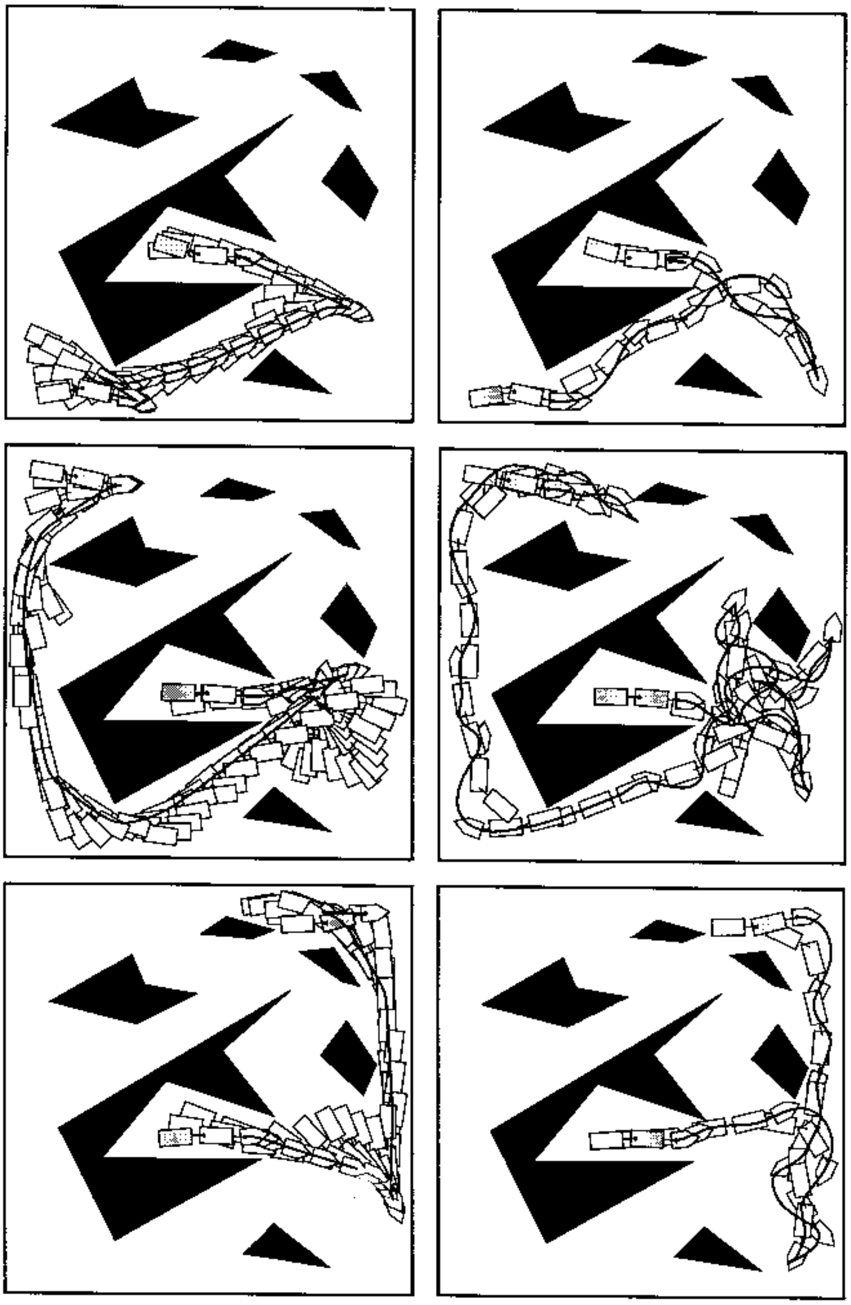


Figure 18: Examples of solutions computed by the multi-level approach (the left column shows the first "semi-holonomic" paths)

5.7 On the computational complexity of nonholonomic path planning

Evaluating the computational complexity of the approaches introduced above is a difficult task. More generally, the complexity of the nonholonomic path planning problem is an open problem.

For small-time controllable systems, we have seen that the existence of a solution is characterized by the existence of any collision-free path for the associated holonomic system. The complexity of deciding whether a solution exists is then equivalent to the complexity of the classical piano mover problem (see [59] for an overview). The complexity for other systems (e.g., with drift) is an open problem.

In this section we give an account of results providing lower bounds on the complexity of nonholonomic paths for small-time controllable mobile robots. By reference to the approximation scheme, we may define the complexity of a collision-free nonholonomic path by the length of the sequence of admissible paths approximating a holonomic one. This definition depends a priori on the steering method used to approximate a holonomic path. A more intrinsic definition consists in considering the approximation scheme that uses Steer_{opt}. Indeed the cost of the optimal paths induces a (nonholonomic) metric in the configuration space. A possible definition of the complexity of a path is the minimum number of balls computed with the nonholonomic metric and covering the path. For instance the complexity of the paths appearing in Figure 15 is 7 in both cases. This definition allows to link the complexity of nonholonomic path planning with the clearance of the free-space.

Let us consider the classical parking task problem illustrated in Figure 19 for a car-like robot. The solutions have been computed by the algorithm presented in Section 5.3.2. The steering method to approximate the holonomic path is Steer_{opt} which computes Reeds&Shepp's shortest paths. The length of the shortest paths induces a metric d_{RS} in configuration space. The shape of the balls computed with this metric appears in Figure 3 (top). Let us consider a configuration $X = (x, y, \theta)$ near the origin O. In Annex B we prove that:

$$\frac{1}{3}(|x| + |y|^{\frac{1}{2}} + |\theta|) \;\leq\; d_{RS}(O, X) \;\leq\; 12(|x| + |y|^{\frac{1}{2}} + |\theta|)$$

As a consequence, the number of balls required to cover the "corridor" where the car has to be parked varies as ϵ^{-2} with ϵ being the width of the corridor. Moreover each elementary shortest path providing a motion in the direction of the wheel axis requires exactly two cusps. Then the number of maneuvers to park a car is in $\Omega(\epsilon^{-2})$.

Such a reasoning may be generalized to small-time controllable systems. Let us consider a control system defined by a set of vector fields; let us

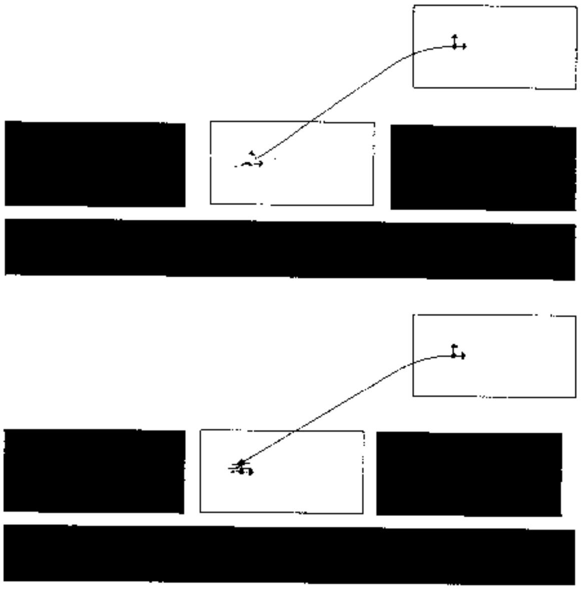


Figure 19: The number of maneuvers varies as the inverse of the square of the free space.

assume that the tangent space at every point can be spanned by a finite family of these vector fields together with their Lie brackets (i.e., the system verifies the LARC at every point). The minimal length of the Lie bracket required to span the tangent space at a point is said to be the degree of nonholonomy of the system at this point.

The cost of the optimal paths induces a metric in the configuration space of the system. A ball of radius r corresponding to this metric is the set of all the points in the configuration space reachable by a path of cost lesser than r. The balls grow faster in the directions given by the vector fields directly controlled than in the directions defined by the Lie brackets of these vector fields. A powerful result from sub-Riemannian geometry shows that the growing law depends on the degree of bracketing (see [9, 40, 126, 75]): when r is small enough, the ball grows as r in the directions directly controlled; it grows as r^d in the directions spanned by Lie brackets of length d.

Figure 20 illustrates this complexity modeling on a mobile robot with two trailers. We have seen in Section 2.3 that the degree of nonholonomy of this system is 4 when $\varphi_1 \neq \frac{\pi}{2}$ (regular points) and 5 everywhere else. This means that the complexity of the parking task is in $\Omega(\epsilon^{-4})$ while the complexity of the exotic example on the right side (the mobile robot can not escape the room...) is in $\Omega(\epsilon^{-5})$. These worst case examples may be generalized to an arbitrary number of trailers: the degree of nonholonomy for a mobile robot with n trailers has been proved to be $n + 2$ at regular points and $Fib(n+3)$ when all the relative angles of the trailers are $\frac{\pi}{2}$ [73, 50] ($Fib(n+3)$ is the $(n+3)$th number of the famous sequence of Fibonacci defined by $Fib(i+2) = Fib(i+1) + Fib(i)$, i.e., 1, 1, 2, 3, 5, 8, 13...). This means that the complexity of the problems appearing in Figure 20 and generalized to n trailers are respectively $\Omega(\epsilon^{-n-2})$ (simply exponential in n) and $\Omega(\epsilon^{-Fib(n+3)})$ (doubly exponential in n).

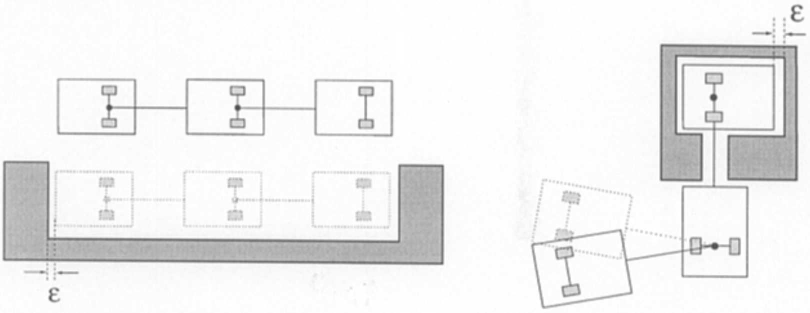


Figure 20: The complexity of admissible paths for a mobile robot with n trailers are respectively $\Omega(\epsilon^{-n-2})$ (case on the left side) and $\Omega(\epsilon^{-Fib(n+3)})$ (case on the right side).

6 Other approaches, other systems

This section overviews other works related to nonholonomic path planning for mobile robots. They deal either with direct approaches based on dynamic programming techniques, or with specific systems.

Combining discrete configuration space and piece-wise constant inputs: Barraquand and Latombe propose in [6, 7] a direct approach to nonholonomic path planning. It applies to car-like robots with trailers. The model of the car corresponds to the control system (4) introduced in Section 2.2.2. Four input types are chosen in $\{-1, 1\} \times \{\zeta_{min}, \zeta_{max}\}$; they correspond to backward or forward motions with an extremal steering angle. The admissible paths are generated by a sequence of these constant inputs, each of them being applied over a fixed interval of time δt. Starting from the initial configuration the search generates a tree: the successors of a given configuration X are obtained by setting the input to one of the four values and integrating the differential system over δt. The configuration space is discretized into an array of cells of equal size (i.e. hyperparallelepipeds). A successor X' of a configuration X is inserted in the search tree if and only if the computed path from X to X' is collision-free and X' does not belong to a cell containing an already generated configuration. The algorithm stops when it generates a configuration belonging to the same cell as the goal (i.e., it does not necessarily reach the goal exactly).

The algorithm is proved to be asymptotically complete w.r.t. to both δt and the size of the cells. As a brute force method, it remains quite time-consuming in practice. Its main interest is that the search is based on Dijkstra's algorithm which allows to take into account optimality criteria such that the path length or the number of reversals. Asymptotical optimality to generate the minimum of reversals is proved for the car-like robot alone.

Progressive constraints: In [34] Ferbach combines the two step approach presented in Section 5.3 and a so-called variational approach. It applies for small-time controllable system. First, a collision-free path is generated. Then the nonholonomic constraints are introduced progressively. At each iteration, a path is generated from the previous one to satisfy more severe nonholonomic constraints. The search explores the neighborhood of the current path according to a dynamic programming procedure. The progressiveness of the search is obtained by taking random tangent vectors chosen in neighborhoods of the admissible ones and by making these neighborhoods decreasing to the set of admissible tangent vectors.

The method is neither complete nor asymptotically complete. Completeness would require back-tracking that would be expensive. Nevertheless simulations have been performed with success for a mobile robot with three trailers and for two tractor-trailer robots sharing the same environment.

Car-like robots moving forward: After the pioneering work of Dubins who characterized the shortest paths for a particle moving with bounded curvature [33], attempts have been done to attack the path planning for car-like robots moving only forward. Except some algorithms that do not verify any general completeness properties (e.g., [62, 122, 128]), they are only few results addressing the general problem. All of them assume that the robot is reduced to a point. In [38], Fortune and Wilfong propose an algorithm running in exponential time and space to decide if a path exists; the algorithm does not generate the solution. Jacobs and Canny's algorithm [46] is a provably good approximation algorithm that generates a sequence of elementary feasible paths linking configurations in contact with the obstacles. According to the resolution of a contact space discretization, the algorithm is proved to compute a path which is as close as possible to the minimal length path. More recent results solve the problem exactly when the obstacles are bounded by curves corresponding to admissible paths (i.e., the so-called moderate obstacles) [2. 16].

Nonholonomic path planning for Dubins' car then remains a difficult and open problem. Barraquand and Latombe's algorithm [6] may be applied to provide an approximated solution of the problem. Finally optimal path planning for Dubins' car has been recently proved to be NP-hard [89].

Multiple mobile robots: Nonholonomic path planning for the coordination of multiple mobiles robots sharing the same environment has been addressed along two main axis: centralized and decentralized approaches[16].

In the centralized approaches the search is performed within the Cartesian product of the configuration spaces of all the robots. While the problem is PSPACE-complete [44], recent results by Svestka and Overmars show that it is possible to design planners which are efficient in practice (until five mobile robots) while being probabilistically complete [117]: the underlying idea of the algorithm is to compute a probabilistic roadmap constituted by elementary (nonholonomic) paths admissible for all the robots considered separately; then the coordination of the robots is performed by exploring the Cartesian product of the roadmaps. The more dense is the initial roadmap, the higher is the probability to find a solution in very cluttered environments.

In [1], Alami reports experiments involving ten mobile robots on the basis of a fully decentralized approach: each robot builds and executes its own plan by merging it into a set of already coordinated plans involving other robots. In such a context, planning is performed in parallel with plan execution. At any time, robots exchange information about their current state and their current paths. Geometric computations provide the required synchronization along the paths. If the approach is not complete (as a decentralized schemes), it is sufficiently well grounded to detect deadlocks. Such deadlocks usually involve only few robots among the fleet; then they may be overcome by applying a centralized approach locally.

7 Conclusion

The algorithmic tools presented in this chapter show that the research in motion planning for mobile robots reaches today a level of maturity that allows their transfer on real platforms facing difficult motion tasks.

Numerous challenging questions remain open at a formal level. First of all, there is no nonholonomic path planner working for any small-time controllable system. The case of the mobile robot with trailers shown in Figure 4 (right) is the simplest canonical example which can conduce new developments. A second issue is path planning for controllable and not small-time controllable systems; Dubins' car appears as another canonical example illustrating the difficulty of the research on nonhonomic systems. Souères–Boissonnat's chapter emphasizes on recent results dealing with the computation of optimal controls for car-like robots; it appears that extending

[16] We refer the reader to [118] for a more detailed overview on this topic.

these tools to simple systems like two-driving wheel mobile robots is today out of reach.

Perhaps the most exciting issues come from practical applications. The motion of the robot should be performed in the physical world. The gap between the world modeling and the real world is critical. Usually, path planning assumes a two-steps approach consisting in planning a path and then executing it via feedback control. This assumption holds under the condition that the geometric model of the environment is accurate and that the robot's Cartesian coordinates are directly and exactly measured. Designing a control law that executes a planed path defined in a robot centered frame may be sufficient in manufacturing applications; it is not when dealing with applications such as mobile robot outdoor navigation for instance. In practice, the geometric model of the world and the localisation of the robot should be often performed through the use of embarked extereoceptive sensors (ultrasonic proximeters, infrared or laser range finder, laser or video cameras...).

Uncertainties and sensor-based motions are certainly the two main keywords to be considered to reach the ultimate objectives of the motion planning. Addressing these issues requires to revisit the motion planning problem statement: the problem is to plan not a robot-centered path but a sequence of sensor-based motions that guaranty the convergence to the goal. The solution is no more given by a simple search in the collision-free configuration space. This way is explored in manufacturing applications for several years; it is difficult in mobile robotics where nonholonomy adds another difficulty degree.

A Nonholonomic systems: combinatorial issues

The material of this annex uses the concepts of a distribution, also known as a Pfaffian system (see for instance [126]), and of the Free Lie Algebra (see [17]).

Let us recall that every Lie operator has to verify skew-symmetry $[f, g] = -[g, f]$ and the Jacobi identity $[f, [g, h]] + [g, [h, f]] + [h, [f, g]] = 0$.

Consider the $(n-r)$-distribution Δ associated with a robotic system. The goal of this annex is to present an *algorithm* for testing the controllability of that system at a point. Precisely, we are interested in the rank of $LA(\Delta)$ (i.e., the distribution spanned by all the combinations of Lie brackets of vector fields in Δ). We can consider a basis $\mathcal{F}$ of Δ together with all the combinations of Lie brackets built upon that basis.

A.1 Filtrations and degree of nonholonomy

To do this, one may consider a brute force strategy consisting in building iteratively the following increasing sequence of distributions: $\Delta_i = \Delta_{i-1} + [\Delta_{i-1}, \Delta_{i-1}]$ where $[\Delta_{i-1}, \Delta_{i-1}]$ is the linear space spanned by all the brackets $[f, g]$ for f and g in Δ_{i-1}. By putting $\Delta = \Delta_1$, the Control Lie Algebra $LA(\Delta)$ is precisely defined as $\bigcup_i \Delta_i$. But in fact, a more efficient strategy can be used. First of all, let us define a parameter estimating the complexity of a combination of Lie brackets. The *degree* of a combination is the number of elements in $\mathcal{F}$ defining the combination. For example the degree of $[., [., [., .]], [., [., .]]]$ is 7. Now, our strategy will consist in building all the brackets of a given degree, step by step. This strategy is founded on the following iterative construction. We denote Δ by Δ_1. Then Δ_i is defined by:

$$\Delta_i = \Delta_{i-1} + \sum_{j+k=i} [\Delta_j, \Delta_k].$$

It verifies:

$$\Delta_1 \subset \Delta_2 \subset \Delta_3 \subset \cdots \subset LA(\Delta) \qquad \text{and} \qquad LA(\Delta) = \bigcup_i \Delta_i.$$

The set of all the Δ_is is called a *filtration* associated with Δ.

Remark: Such a construction can be viewed as a "breadth-first" construction. Some authors [126, 123] use another construction. $\tilde{\Delta}$ is denoted by $\tilde{\Delta}_1$. Then $\tilde{\Delta}_i$ is defined by:

$$\tilde{\Delta}_i = \tilde{\Delta}_{i-1} + [\tilde{\Delta}_1, \tilde{\Delta}_{i-1}].$$

Again:

$$\tilde{\Delta}_1 \subset \tilde{\Delta}_2 \subset \tilde{\Delta}_3 \subset \cdots \subset LA(\Delta).$$

Such a construction can be viewed as a "depth-first" construction. Using skew-symmetry and the Jacobi identity, we may verify that both constructions are the same[17]. We will prefer the first presentation that corresponds exactly to the concept of Phillip Hall families introduced below.

On the other hand, at a point X of our manifold (the configuration space), $(n-r) \leq \text{rank}\, \Delta_i(X) \leq n$ [18]. Moreover, if $\Delta_i(X) \neq \Delta_{i-1}(X)$, then $\text{rank}\, \Delta_i(X) > \text{rank}\, \Delta_{i-1}(X)$. Hence, if we consider the construction *locally* (i.e., by applying the distributions at a point), we can conclude that there exists an index p_c such that $\Delta_{p_X-1}(X) \neq \Delta_{p_X}(X) = \Delta_{p_X+1}(X) = \cdots$. The construction always stabilizes. The index p_X is the *degree of nonholonomy* of the system at X. Therefore a system is controllable at X if and only if $\text{rank}\, \Delta_{p_X} = n$ (if $p_X = 1$ we are locally in the holonomic situation). Notice that, from a global point of view, this stabilization property is not true, since the degree of nonholonomy may change from point to point. A close analysis of possible singularities shows that this degree may be arbitrarily high at singular points—even when we start with a regular distribution, the filtration we build may acquire some weird singularities. So, the degree of nonholonomy may be unbounded when X varies.

Remark: It is possible to define a *global* degree of nonholonomy of a nonholonomic system, as the maximum of pointwise degrees of nonholonomy. There are no obvious applications of this notion. Also, keep in mind that this global degree can be infinite, though it will stay bounded in the particular cases we consider.

A.2 A controllability algorithm

In this section we define an algorithm for testing the controllability of a given system at a point based upon the previous construction. We have to use a basis $\mathcal{F}$ of Δ. According to that construction, we build:

$$\begin{aligned} \mathcal{F}_1 &= \mathcal{F} \\ \mathcal{F}_i &= \mathcal{F}_{i-1} \bigcup_{j+k=i} [\mathcal{F}_j, \mathcal{F}_k] \end{aligned}$$

where, now, $[\mathcal{F}_j, \mathcal{F}_k]$ is no longer viewed as a linear space, but as a *finite* family of brackets. Each $\mathcal{F}_i$ contains of course a basis of Δ_i. Again, we can define the union $\mathcal{LA}(\mathcal{F})$ of all theses families and we have:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \cdots \subset \mathcal{LA}(\mathcal{F})$$

[17] For example, take $[[f,g],[f,[f,h]]]$, an element of Δ_5:

$$[[f,g],[f,[f,h]]] \;=\; -[f,[f,[g,[f,h]]]]+[f,[g,[f,[f,h]]]]+[f,[h,[f,[f,g]]]]-[h,[f,[f,[f,g]]]].$$

Hence, it belongs to $\tilde{\Delta}_5$ too.

[18] We denote by $\Delta_i(X)$ the linear subspace of the tangent space in X, obtained by applying the distribution Δ_i at X.

This is clearly an infinite family, but, during the real process, we can check out the added elements if they happen to be linearly dependent on the previous ones.

Even if we know only about the relations pertinent to the concept of a Lie Algebra, we can take advantage of these to compute only relevant elements of what is called the *Free Lie Algebra.*

A.2.1 Phillip Hall families

In this section[19] the elements of $\mathcal{LA}(\mathcal{F})$ are considered as formal expressions produced by the construction above, i.e., they are not actually evaluated as vector fields belonging to a distribution. From this point of view, $\mathcal{LA}(\mathcal{F})$ is considered as a *Free Lie Algebra.* Our current problem is to enumerate a basis of this algebra, i.e., to get rid of redundant elements using only skew-symmetry and the Jacobi identity. Such a basis can be found via a Phillip Hall family.

The degree of an element f in $\mathcal{LA}(\mathcal{F})$ is denoted by degree(f): this is the degree of the monomial defining f [20]. According to our notations, a *Phillip Hall family* (PH-family for short) of $\mathcal{LA}(\mathcal{F})$, is any totally ordered subset $(\mathcal{PH}, \prec)$ such that:

- If $f \in \mathcal{PH}$, $g \in \mathcal{PH}$ and degree(f) $<$ degree(g) then $f \prec g$;
- $\mathcal{F} \subset \mathcal{PH}$;
- $\mathcal{PH} \bigcap \mathcal{F}_2 = \{[f,g], \quad f \prec g\}$;
- An element $f \in \mathcal{LA}(\mathcal{F})$ with degree(f) ≥ 3 belongs to $\mathcal{PH}$ if and only if $f = [u,[v,w]]$ with u, v, w in $\mathcal{PH}$, $[v,w]$ in $\mathcal{PH}$, $v \preceq u \prec [v,w]$ and $v \prec w$.

The main property of a PH-family is that, taking skew-symmetry and the Jacobi identity into account, it yields a *basis* of the free Lie algebra $\mathcal{LA}(\mathcal{F})$ [17].

The proof of existence of such a family is easy; it is an iterative one. In the context of our control problem, it can be extended into the following algorithm.

[19] The material used in this section comes from [17]. We want just to give a rough idea of the concept and of its pertinence with respect to our problem. Interested readers will find a more rigorous presentation in this reference.

[20] We use the word "degree" with two different meanings, according to whether we speak of a bracket or of a nonholonomic systLu.. This may introduce some confusion, but both terms are already used in the literature (see for instance [123]).

A.2.2 The algorithm

The idea is to build a PH-family, based upon a graded family of sets $\mathcal{H}_i$, where $\mathcal{H}_i$ is a part of $\mathcal{F}_i$. We will also build a total order $\prec$ on the union $\mathcal{H}_i$. Assume first that $\mathcal{F}$ is totally ordered by $<$ and set $\mathcal{H}_1 = \mathcal{F}$. The order $\prec$ on $\mathcal{H}_1$ is the same as the order $<$ on $\mathcal{F}$. The next set $\mathcal{H}_2$ is defined as the set of all $[f, g]$, with f, g elements of $\mathcal{H}_1$ and $f \prec g$. Endow $\mathcal{H}_2$ with any total order, and define $\prec$ on $\mathcal{H}_1 \bigcup \mathcal{H}_2$ by setting $f \prec g$, for f in $\mathcal{H}_1$ and g in $\mathcal{H}_2$.

The rest of the algorithm consists in building the sets $\mathcal{H}_i$ iteratively. Suppose the family $\mathcal{H}_1, \mathcal{H}_2, \ \ldots \ , \mathcal{H}_{i-1}$ is given. Denote it as $\mathcal{H}$. Define $\mathcal{H}_i$ according to the definition of a PH-family. That is, $\mathcal{H}_i$ is the set of all $f = [u, [v, w]]$ verifying: $u \in \mathcal{H}_j$, $[v, w] \in \mathcal{H}_{i-j}$, $v \preceq u \prec [v, w]$ and $v \prec w$. Choose a total order on $\mathcal{H}_i$ and extend it to $\mathcal{H} \bigcup \mathcal{H}_i$: $f \prec g$ if $f \in \mathcal{H}$ and $g \in \mathcal{H}_i$. It is almost obvious that the family $\mathcal{H}_i$ is a PH-family and, furthermore, that the degree of an element of $\mathcal{H}_i$ is precisely i.

We can use this construction to design an algorithm for testing the controllability of a system at a point X of the manifold. Our algorithm adds new brackets to the PH-family step by step, but now, we check further the value of each new bracket as a potential member of a basis at X. If we ultimately obtain a basis, the system is controllable at X.

In the procedure appearing in Figure 21, $\mathcal{B}$ denotes the free family that will eventually become a basis, *cnt* is the current number of element of that basis. The initial distribution is $(n-r)$-dimensional at the point X. For an order on $\mathcal{H}$, we assume that we have an initial order on $\mathcal{F}$; then we simply take the order of chronological computation. Finally $\lfloor x \rfloor$ is the integer part (floor function) of the real x.

One can verify that the procedure builds sets $\mathcal{H}_i$ defining a PH-family. Therefore, it appears clearly that the system is controllable at X if and only if *Controllability*(X) terminates. This also means that the procedure never stops otherwise.

Example Part 1: For a classical example [17, 54], take $\mathcal{F} = \{f_1, f_2\}$. The first 14 elements of the PH-family generated by the procedure (if it does not stop before) are:

$$\begin{array}{lll}
\mathcal{H}_1 : & f_1 & f_2 \\
\mathcal{H}_2 : & f_3 = [f_1, f_2] & \\
\mathcal{H}_3 : & f_4 = [f_1, [f_1, f_2]] & f_5 = [f_2, [f_1, f_2]] \\
\mathcal{H}_4 : & f_6 = [f_1, [f_1, [f_1, f_2]]] & f_7 = [f_2, [f_1, [f_1, f_2]]] \\
 & f_8 = [f_2, [f_2, [f_1, f_2]]] & \\
\mathcal{H}_5 : & f_9 = [f_1, [f_1, [f_1, [f_1, f_2]]]] & f_{10} = [f_2, [f_1, [f_1, [f_1, f_2]]]] \\
 & f_{11} = [f_2, [f_2, [f_1, [f_1, f_2]]]] & f_{12} = [f_2, [f_2, [f_2, [f_1, f_2]]]] \\
 & f_{13} = [[f_1, f_2], [f_1, [f_1, f_2]]] & f_{14} = [[f_1, f_2], [f_2, [f_1, f_2]]]
\end{array}$$

```
Procedure $Controllability(X)$
    (initialize $\mathcal{H}_1$)
    $\mathcal{H}_1 \leftarrow \mathcal{F}$
    $\mathcal{B} \leftarrow \mathcal{F}$
    $cnt \leftarrow n - r$
    (build $\mathcal{H}_2$)
    For $f$, $g$ in $\mathcal{H}_1$, $f \prec g$ do
        add $[f, g]$ in $\mathcal{H}_2$;
        If $\{[f, g](X)\} \bigcup \mathcal{B}(X)$ is a free family $^{(1)}$
          then
                add $[f, g]$ in $\mathcal{B}$
                $cnt \leftarrow cnt + 1$
    $i \leftarrow 2$
    While $cnt < n$ do
        $i \leftarrow i + 1$
        (build $\mathcal{H}_i$)
        For $1 \leq j \leq \lfloor i/2 \rfloor$ do
            For $f \in \mathcal{H}_j$, $g = [u, v] \in \mathcal{H}_{i-j}$ do
                If $u \preceq f$ $^{(2)}$
                  then
                        add $[f, g]$ in $\mathcal{H}_i$
                        If $\{[f, g](X)\} \bigcup \mathcal{B}(X)$ is a free family $^{(3)}$
                          then
                                add $[f, g]$ in $\mathcal{B}$
                                $cnt \leftarrow cnt + 1$
```

Figure 21: An algorithm to check controllability

Example Part 2: Consider now a 3-body mobile robot (i.e. the classical two degree of freedom mobile robot with two trailers shown in Figure 4, left). The configuration space is a 5-dimensional manifold. Let X be a point of coordinates $(x, y, \theta, \varphi_1, \varphi_2)$. The rolling constraints of the three bodies provide 3 nonholonomic links. We can prove (see [7] [65] for details) that the 2-dimensional associated distribution has the following vector fields as basis [21]:

$$f_1 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\sin\varphi_1 \\ \sin\varphi_1 - \cos\varphi_1 \sin\varphi_2 \end{pmatrix} \qquad f_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

The first elements of a PH-family are displayed in Example Part 1. We can verify that the algorithm stops with $\{f_1, f_2, f_3, f_4, f_6\}$ as a basis for every point X verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. The algorithm stops with $\{f_1, f_2, f_3, f_4, f_9\}$ for the remaining hyperplane[22].

Remark: Finally, the rank condition holds everywhere and we can conclude that the corresponding system is controllable.

In this example, notice that the algorithm checks $6 - 2 = 4$ "candidates" in the first case, and $9 - 2 = 7$ in the second one. What happens in the general case ?

The core of the algorithm is the construction of a PH-family. The dimension n of the manifold being a constant integer of our problem, the only tests needing a subroutine depending on n are (1) and (3). Their complexity is asymptotically negligible. Therefore the worst-case complexity of the algorithm is dominated by the complexity of building a PH-family. The relevant parameter is the value of i when the algorithm stops. Because of the test (2), our procedure for building a PH-family is not optimal. But, here, we just want to find the minimal complexity of any algorithm that builds a PH-family. Now, the complexity of computing all the elements of a set $\mathcal{H}_i$ is bounded below by the number of all the elements in $\mathcal{H}_j$, $j \leq i$, and it has been proven that this number is

$$\sum_{1 \leq j \leq i} \alpha j \qquad \text{with} \qquad \alpha(j) = \frac{1}{j} \sum_{d|j} \mu(d)(n-r)^{\frac{j}{d}}$$

[21] This distribution is computed without reference to any control system. It is built just from the non-sliding hypothesis applied to each body.

[22] More precisely: $f_5 = f_1$, $\det\{f_1, f_2, f_3, f_4, f_6\} = -\cos(\varphi_1)$, $f_7 = 0$, $f_8 = -f_3$ and finally $\det\{f_1, f_2, f_3, f_4, f_9\} = -1 - \cos^2(\varphi_1)\cos(\varphi_2)$.

where μ designates the Möbius function:

$$\begin{array}{rcl} \mu \ :\ \mathbf{N}^* & \longrightarrow & \{-1,0,1\} \\ m & \longmapsto & \mu(m) = \begin{cases} 0 & \text{if } m \text{ is the square of a prime integer} \\ (-1)^k & \text{otherwise, where } k \text{ is the number} \\ & \text{of primes dividing } m. \end{cases} \end{array}$$

For example, setting $(n-r) = m$, we have $\alpha(1) = m$, $\alpha(2) = \frac{1}{2}(m^2 - m)$, $\alpha(3) = \frac{1}{3}(m^3 - m)$, $\alpha(4) = \frac{1}{4}(m^4 - m^2)$, $\alpha(5) = \frac{1}{5}(m^5 - m)$, $\alpha(6) = \frac{1}{6}(m^6 - m^3 - m^2 + m)$. One may verify the first 5 values on the current example.

If the algorithm runs for a point X and stops with a family $\mathcal{H}_i$, the system is said to be *completely nonholonomic* at X (i.e., all missing dimensions can be recovered; its degree of nonholonomy is maximal; it is controllable). Besides, its degree of nonholonomy at X is i.

We have to prove this latter result. Indeed, the algorithm above clearly depends upon the basis $\mathcal{F}$ we chose for the distribution Δ. However, the concept of degree of nonholonomy does not. Now, it is a general result from the Lie Algebra Theory that $\bigcup_{j \leq i} \mathcal{H}_j$ constitutes a basis of the nilpotent free Lie algebra $\mathcal{L}\mathcal{A}_i(\mathcal{F})$ defined by taking all the brackets of degrees less than i and by killing all the brackets of greater degrees. See [17] for details. Therefore, i does not depend on our choice of a basis $\mathcal{F}$ of Δ. It truly is the degree of nonholonomy that has been previously defined.

Example Part 3: The degree of nonholonomy of the 3-body mobile robot is 4 at points whose coordinates $(x, y, \theta, \varphi_1, \varphi_2)$ verify $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. It is 5 elsewhere.

Summing up the results of this section: the method we use for testing the controllability of a nonholonomic system at a point is *at least exponential* in the degree of nonholonomy at this point.

Next two sections present classes of systems for which one can stop the procedure *Controllability* after a finite and *known* number of steps.

A.3 Growth vector, regular and singular points

Let us define the growth vector of a controllable nonholonomic system at a point. This concept appears in [126].

Suppose that the distribution associated to our system is $(n-r)$-dimensional. Consider a point X and its degree of nonholonomy q_X. The *growth vector at* X is defined as the sequence $(n_1, \ldots, n_{q_X})$, where $n_1 = (n-r) \leq n_2 \leq \cdots \leq n_{q_X} = n$ and n_i is the dimension at X of the linear space generated by combinations of brackets of degree less than i.

Example Part 4: Let us recall that $\{f_1, f_2, f_3, f_4, f_6\}$ constitutes a basis for points whose coordinates $(x, y, \theta, \varphi_1, \varphi_2)$ verify $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$, while

$\{f_1, f_2, f_3, f_4, f_9\}$ works elsewhere. One can verify (by computing the dimension of the linear spaces for each level) that the growth vector at points verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$ is $(2, 3, 4, 5)$, while it is $(2, 3, 4, 4, 5)$ elsewhere.

All the above tools work only locally. For instance, we have just seen that the growth vector is not the same everywhere in the manifold. The global viewpoint is not easy to reach. A first step is to study what happens in a neighborhood of a point.

A filtration $\{\Delta_i\}$ is *regular* at a point X if the growth vector is constant in a neighborhood of X [126] [123]. This means that all the ranks of $\Delta_i(.)$ are constant in the neighborhood. Otherwise, the filtration is *singular* and the corresponding point is a *singularity*. Most of the problems we encounter when we try to define growth vectors and degrees of nonholonomy derive from the presence of singularities. By extension, we will say that a system is *regular* if the corresponding filtration is regular everywhere.

Example Part 5: The 3-body system is not regular; more precisely the corresponding filtration is regular at points verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. It is singular at the remaining points. Remark that the growth vector is strictly increasing for regular points.

For regular systems, the degree of nonholonomy is a constant. It can also be shown (see [111]) that the growth vector is strictly increasing, so the procedure we designed always stops in that particular case.

A.4 Nilpotent and nilpotentizable systems

Now, let us consider the following case.

Suppose that all the Lie brackets of degree greater than k vanish. In this case, the sequence $\mathcal{F}_i$ stabilizes:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots \subset \mathcal{F}_k = \mathcal{LA}(\mathcal{F}).$$

Such systems are called *nilpotent of order k* (see [17] for a general definition of the concept in the Lie Algebra framework).

For nilpotent systems of order k, we can stop the procedure *Controllability* as soon as all the Lie brackets of degree k or less are generated. If the procedure does not yield a basis, then the system is not controllable.

Example Part 6: In our example, we may verify that $[f_2, [f_2, f_1]] = -f_1$. Set $ad_f(g) = [f, g]$. Then[23]: $ad_{f_2}{}^{2m}(f_1) = (-1)^m f_1$. The system is not nilpotent.

In some cases, a non-nilpotent system can be transformed into a nilpotent one via a linear change of controls called a *feedback transformation*. Quite

[23] This example appears in [54] for the unicycle and the car-like robot, i.e., systems equivalent to our current system without trailers.

logically, such systems are called *feedback nilpotentizable*. [54] gives some examples of feedback nilpotentizations (e.g., the unicycle, a car-like system and a car-like system with a trailer). See also [43] for sufficient conditions for a system to be nilpotentizable.

A.5 Well-controllability

At this stage of the presentation, let us return to the planning problem. This section introduces the concept of well-controllability. As the regularity concept, it deals with the existence of singularities, but this is a more global one.

As we have seen, a general idea for devising a nonholonomic motion planner for controllable systems is to define a procedure that searchs for an admissible collision-free path, taking any collision-free path as a seed for the search[24].

Lafferiere and Sussmann proved that this principle is a general one. A collision-free path is first computed without taking the nonholonomic constraints into account. Lafferiere and Sussmann's method [54] roughly consists of expressing the first holonomic path into some "local coordinate system" (see Section 4); from these coordinates, because the system is controllable, the authors show that it is possible to explicitely define an admissible control (and then an admissible path) that locally steers the system from a given point (on the first path) to any other on the first path inside a given neighborhood. Because the planner has to work *a priori* everywhere, one has to define a procedure that guarantees to find a local coordinate system *everywhere*. The existence of such a coordinate system is a technical point essential for the method. It is solved by considering an *extended system* associated with the original one; this new system is obtained by adding virtual controls working on vector fields defined from a PH-family of the original system. Since the nonholonomic distribution Δ is $(n-k)$-dimensional, it seems *a priori* that k additional controls would suffice to make the system holonomic. In fact, in order to avoid singularities (understood as points where the transformation matrix would be non invertible), one has generally to add more controls. Lafferriere and Sussmann note also that additional controls make easier the choice of a transformation matrix with a good condition number.

Let us illustrate this point using our example.

Example Part 7: Recalling Example Part 2, a local coordinate system defined from $\{f_1, f_2, f_3, f_4, f_6\}$ will encounter singularities. Following Lafferiere and Sussmann's method, a possible extended control system is defined

[24] [61] pinpointed this method for the car-like robot, while [63] presents a first planner using this principle for this case.

by $\{f_1, f_2, f_3, f_4, f_6, f_9\}$; in the process, four controls are added to the original ones. The previous results show that it is everywhere possible to choose in this family a basis that spans $\mathbf{R}^5$.

Now, consider the following family:

$$\begin{array}{lll} & u_0 = f_1 & v_0 = f_2 \\ g_1 = [f_1, f_2] & u_1 = \cos\varphi_1 f_1 + \sin\varphi_1 g_1 & v_1 = \sin\varphi_1 f_1 - \cos\varphi_1 g_1 \\ g_2 = [u_1, v_1] & u_2 = \cos\varphi_2 u_1 + \sin\varphi_2 g_2 & v_2 = \sin\varphi_2 u_1 - \cos\varphi_2 g_2 \\ g_3 = [u_2, v_2] & & \end{array}$$

It is easy to check that the determinant of $\{v_0, v_1, v_2, u_2, g_3\}$ is equal to 1. Therefore $\{v_0, v_1, v_2, u_2, g_3\}$ spans $\mathbf{R}^5$ *everywhere*[25]. According to the previous comments, we can define a *minimal* extended system that never meets with any singularity. Moreover, the transformation matrix has a good condition number. We introduce the concept of a *well-controllable* system.

Definition 4 *An n-dimensional nonholonomic system defined by a distribution Δ is* well-controllable, *if there exists a basis of n vectors fields in the Control Lie Algebra $LA(\Delta)$ such that the determinant of the basis is constant.*

Obviously well-controllability implies controllability. The converse does not hold. Indeed, as we mentionned above, a system can be controllable while the local degrees of nonholonomy are unbounded. This means that the filtration $\{\Delta_i\}$ stabilizes locally, but not globally. In this case, it is impossible to define a basis verifying the conditions of our definition.

The well-controllability concept is a global one and it is related to the planning problem. Indeed, for well-controllable systems, the same "local" coordinate system can work everywhere. This simplifies Lafferiere and Sussmann's planning method. But, though we have a general procedure for testing controllability, we have no general procedure for testing well-controllability. For instance, there is no obvious argument leading to reducing the search of a good basis to a small family, like a PH-Hall family.

Let $\mathcal{B}$ be a basis of the control Lie Algebra verifying the conditions of Definition 4. The *degree of well-controllability* of the system is the maximum degree of all the elements of $\mathcal{B}$.

Remark: If the system is well-controllable, it is obvious that the global degree of nonholonomy is finite.

Example Part 8: The 3-body mobile robot is a well-controllable nonholonomic system. Its degree of nonholonomy is 5, while its degree of well-controllability is at most 8. [66] gives a proof in the general case of a n-body

[25] Be careful: the degree of g_3 equals 8, when it is viewed as a polynomial function with indeterminates f_1 and f_2 in $\mathcal{LA}(\{f_1, f_2\})$.

mobile robot and shows that the degree of well-controllability is at most 2^n in this case.

A.6 An example with unbounded degree of nonholonomy

Finally, our last example doesn't model any robotics systems whatsoever[26]. Rather, it tries to capture the flavor of problems encountered in practice without any tedious computations: for instance, the example of the n-body system is quite similar to this one. Let us work in Euclidean space (x, y, z). Choose an integer n and consider the following vector fields:

$$f = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad g_0 = \begin{pmatrix} 1 + z^n \\ 1 + z^{2n} \\ 0 \end{pmatrix}$$

and the distribution they engender. Computing the associated filtration obviously sums up to computing

$$g_k = \begin{pmatrix} (z^n)^{(k)} \\ (z^{2n})^{(k)} \\ 0 \end{pmatrix}.$$

Now, this system is regular almost everywhere but not everywhere: the point $(0, 0, 0)$ is a major inconvenience. At that point, for any k less than n, the vector field g_k vanishes altogether, so that the growth vector at $(0, 0, 0)$ is $(2, 2, \ldots, 3)$, giving thus a generic case where the nonholonomy degree is arbitrarily high. Furthermore, using standard techniques (Partitions of Unity), it is easy to piece together a denumerable infinity of such singular patches, so that the resulting distribution has an unbounded degree.

This is the typical case where our algorithm will not terminate.

[26] This example was suggested by Marc Espie.

B Self-contained proof of controllability for a car-like robot. Nonholonomic metric approximation

This annex presents two proofs of the controllability of the car-like system. The first one can be read without reference to differential geometric control theory. Then we present a second version of the proof, which illustrates the general proof of controllability theorem given by the Lie algebra rank condition. Finally we provide an estimation of the nonholonomic metric induced by the length of the shortest paths.

B.1 Direct proof

Proposition 1 *The car like system is controllable.*

Proof 1 *It suffices to prove that the system is locally controllable from the origin. Let $X = (x, y, \theta)$ be a point near the origin. The idea of the proof is to build a path consisting of three pieces γ_1, γ_2 and γ_3 attaining from X, in succession, the points $X_1 = (x_1, y_1, 0)$, $X_2 = (x_2, 0, 0)$ and the origin. Clearly, this path can be followed by the car-like system in the reverse direction to go from the origin to X.*

Let γ_1 be the arc of the circle[27] *tangent at X, with length $|\theta|$ following in the direction of $-sign(\theta)$. We assume that $\theta \geq 0$. γ_1 attains the point $X_1 = (x - \sin\theta, y - (1 - \cos\theta), 0)$. We assume that $y - (1 - \cos\theta) > 0$. Other cases would be processed in the same way (see Figure 22).*

Now, let $\gamma_2(\tau)$ be the path consisting of four pieces of same length τ : a forward motion on a straight line segment, a forward motion on an arc of a circle, a backward motion on a straight line and then a backward motion on an arc of a circle. The coordinates of the point attained by this sequence are $(x_1 + \tau - \tau\cos\tau, y_1 - \tau\sin\tau, 0)$. Choose τ_c such that

$$|y - (1 - \cos\theta)| = \tau_c \sin\tau_c \tag{12}$$

Such τ_c always exists (and is unique) for any X sufficiently near the origin. The coordinates of point X_2 attained by $\gamma(\tau_c)$ are : $(x - \sin\theta + \tau_c(1 - \cos\tau_c), 0, 0)$.

Finally, let γ_3 be the straight line motion from X_2 to the origin.

The path formed by the sequence γ_1, γ_2 and γ_3, followed in the reverse direction, goes from the origin to X. Thus the car-like system is locally controllable.

The length of the path built in the proof above is

$$l = |\theta| + 4\tau_c + |x - \sin\theta + \tau_c(1 - \cos\tau_c)| \tag{13}$$

[27] All the arcs of circles used in this proof have the minimum radius of curvature which is assumed to be 1.

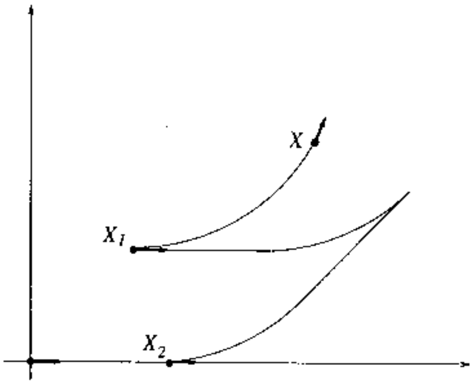


Figure 22: A way to reach the origin

The proof of Proposition 2 below will give an upper bound for this length. This means that, for any time T there exists a neighborhood of the origin such that all the points in this neighborhood can be reached from the origin in time bounded by T. This is the hard core of the small-time controllability concept.

B.2 Proof using Campbell-Haussdorf-Baker-Dynkin formula

This section presents another proof of the previous proposition in order to illustrate the application of Campbell-Haussdorf-Baker-Dynkin formula introduced in Section 4.1.

Proof 2 *Let us consider the following vector fields f_1 and f_2:*

$$f_1 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} \quad , \quad f_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The coordinates of $f_1 + f_2$ and $[f_1, f_1 + f_2]$ are respectively :

$$f_1 + f_2 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 1 \end{pmatrix} \qquad [f_1, f_1 + f_2] = \begin{pmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{pmatrix}$$

We check easily that $\{f_1,\ [f_1, f_1 + f_2],\ f_1 + f_2,\}$ spans the tangent space at every point. Let us prove that the system is locally controllable from the origin. The same proof would hold for every point. Notice that the three vector fields the subpaths γ_3 and γ_1 used in the previous proof correspond to the integration of the flows along f_1 and $f_1 + f_2$ respectively. The path γ_2 is in fact an approximation of the path corresponding to the integration of flow along the vector field $[f_1, f_1 + f_2]$.

Consider a point X in a neighborhood of the origin. Let (t_1, t_2, t_3) be its coordinates in the coordinate system induced by the basis $\{f_1,\ [f_1, f_1 + f_2],\ f_1 + f_2,\}$ which spans the tangent space at the origin. X is reachable from the origin by the following flow :

$$\varphi(t_1, t_2, t_3) = e^{t_1 f_1}\ e^{t_2[f_1, f_1+f_2]}\ e^{t_3(f_1+f_2)} \tag{14}$$

The first and third flows clearly obey the constraints on the controls, but what about the second? To answer this, we consider the following approximation (for $t > 0$) given by the Campbell-Baker-Hausdorff-Dynkin formula.

$$e^{(t[f_1, f_1+f_2]+O(t^{\frac{3}{2}}))} = e^{t^{\frac{1}{2}} f_1} e^{t^{\frac{1}{2}}(f_1+f_2)} e^{-t^{\frac{1}{2}} f_1} e^{-t^{\frac{1}{2}}(f_1+f_2)} \tag{15}$$

This shows that any configuration we can attain by following the flow given in Equation 14, can be approximated by the following flow, which obeys the constraints :

$$\tilde{\varphi}(t_1, t_2, t_3) = e^{t_1 f_1}\ e^{t_2^{\frac{1}{2}} f_1}\ e^{t_2^{\frac{1}{2}}(f_1+f_2)}\ e^{-t_2^{\frac{1}{2}} f_1}\ e^{-t_2^{\frac{1}{2}}(f_1+f_2)}\ e^{t_3(f_1+f_2)} \tag{16}$$

Now, because the mapping $\tilde{\varphi}$ is a local homeomorphism, the inverse image of a neighborhood of the origin in the configuration space is a neighborhood of $(0, 0, 0,)$ in $\mathbf{R}^3$, i.e. a choice exists for t_1, t_2 and t_3 which exactly attains any given configuration in a neighborhood of the origin. Hence the system is small-time controllable at the origin.

In fact, for the car-like robot, the differential equations are sufficiently simple that they can be integrated in closed form. This is far from being true in general. More precisely the flow $\tilde{\varphi}$ attains a point of coordinates :

$$\begin{aligned} x &= t_1 + \mathrm{sign}(t_2)\sqrt{|t_2|}(1 - \cos(\sqrt{|t_2|})) + \sin(t_3) \\ y &= -\mathrm{sign}(t_2)\sqrt{|t_2|}\sin(\sqrt{|t_2|}) + 1 - \cos(t_3) \\ \theta &= t_3 \end{aligned} \tag{17}$$

By expressing this point in the basis $\{f_1,\ [f_1, f_1 + f_2],\ f_1 + f_2,\}$ we obtain the asymptotic estimates $(t_1 + O(t_2^{\frac{3}{2}}),\ t_3,\ t_2 + O(t_2^2))$.

Now, by considering t_1, t_2 and t_3 as time parameters, the proof above shows that the car-like system is small-time controllable.

B.3 Nonholonomic metric approximation

Let us denote by d_{RS} the metric induced by the length of the shortest paths between two configurations.

Proposition 2 *For any point $X = (x, y, \theta)$ sufficiently near the origin $o = (0, 0, 0)$,*

$$\frac{1}{3}(|x| + |y|^{\frac{1}{2}} + |\theta|) \leq d_{RS}(X, o) \leq 12(|x| + |y|^{\frac{1}{2}} + |\theta|)$$

Proof 3 1/ *We first prove the left part of the inequalities. Let us consider any admissible path γ parametrized by arc length in $\mathbf{R}^2$ such that $\gamma(0) = o$. Let us denote by $X(t)$ the point with coordinates $(x(t), y(t), \theta(t))$ attained by an admissible path of length t. Since the path is admissible, then, for each $\tau \in [0, t]$, $\dot{\gamma}(\tau)$ is a linear combination of the vector fields f_1 and $f_1 + f_2$, i.e. : $\dot{\gamma}(\tau) = \lambda(\tau)f_1(\tau) + \mu(\tau)(f_1(\tau) + f_2(\tau))$, with $|\lambda(\tau)| \leq 1$ and $|\mu(\tau)| \leq 1$. Then :*

$$|y(t)| \leq \int_0^t |\dot{y}(\tau)|d\tau = \int_0^t |(\lambda(\tau) + \mu(\tau))\sin\theta(\tau)|d\tau \leq 2\int_0^t |\sin\theta(\tau)|d\tau.$$

Now, we may write that $|\sin\theta(\tau)| \leq |\theta(\tau)|$. Moreover, due to the bound on the curvature, the variation of the orientation $\theta(\tau)$ is less than the length of the path from the origin to $X(\tau)$. Hence :

$$\int_0^t |\sin\theta(\tau)|d\tau \leq \int_0^t \tau d\tau$$

Finally :

$$|y(t)| \leq 2\int_0^t \tau d\tau = t^2 \tag{18}$$

Since this relation holds for any admissible path, it holds for the shortest one. This means that, for any point $X = (x, y, \theta)$ sufficiently near the origin :

$$|y|^{\frac{1}{2}} \leq d_{RS}(X, o) \tag{19}$$

Now, the following inequalities hold :

$$|x| \leq d_{RS}(X, o) \quad and \quad |\theta| \leq d_{RS}(X, o) \tag{20}$$

Then by adding inequalities (19) and (20), we obtain the following lower bound :

$$\frac{1}{3}(|x| + |y|^{\frac{1}{2}} + |\theta|) \leq d_{RS}(X, o)$$

2/ *The second part of the proof deals with the right side of the inequalities. We consider the path built in Proof 1 of Property 1 starting from a point $X = (x, y, \theta)$ near the origin to the origin. The length l of this path is (see expression (13) above) :*

$$l \;=\; |\theta| + 4\tau_c + |x - \sin\theta + \tau_c(1 - \cos\tau_c)|$$

with τ_c satisfying equation (12). A first upper bound of l is clearly done by :

$$l \leq |\theta| + 4\tau_c + |x| + |\sin\theta| + |\tau_c(1-\cos\tau_c)|$$

By using the inequalities $|\sin\theta| \leq |\theta|$ and $1-\cos\tau_c \leq 1$ we obtain a second upper bound :

$$l \leq 2|\theta| + 5\tau_c + |x|$$

Now, let us consider the equation (12) : $|y-(1-\cos\theta)| = \tau_c \sin\tau_c$. For X sufficiently near the origin, τ_c is sufficiently small for verifying $\sin\tau_c \geq \frac{1}{4}\tau_c$. Then we may write $\tau_c \leq 2|y-(1-\cos\theta)|^{\frac{1}{2}}$. Moreover $|1-\cos\theta| \leq \theta^2$ for θ sufficiently small. Finally,

$$\begin{aligned} l &\leq |x| + 10|y|^{\frac{1}{2}} + 10|1-\cos\theta|^{\frac{1}{2}} + 2|\theta| \\ &\leq |x| + 10|y|^{\frac{1}{2}} + 12|\theta| \\ &\leq 12(|x| + |y|^{\frac{1}{2}} + |\theta|) \end{aligned}$$

This upper bound holds for the initial path we considered. It then holds for the shortest path. This concludes the proof.

C Shortest paths and obstacle distance for car-like robots

In the framework of the researches in nonholonomic motion planning for mobile robots, the case of the car-like robot has been the most investigated one (see e.g. [67] and the references therein). Numerous results are based on the knowledge of the shortest paths.

C.1 History of the shortest paths for a car-like robot.

A car-like robot, the position and direction of which are defined by the coordinates (x, y) of the reference point and the angle θ between the abscissa axis and the main axis of the car (see Fig. 2), is completely specified as a point (x, y, θ) in the configuration space $\mathbf{R}^2 \times S^1$. Assuming the linear velocity constant, we have seen in Section 2.2 that the motion is defined by the following control system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \qquad (21)$$

where $|v(t)| = 1$ and $|\omega(t)| \leq 1$ are, respectively, the linear and angular velocity of the car. The lower bound on the turning radius is supposed to be 1. This model corresponds to the motion in the plane of a particle subject to curvature constraints.

The study of the shortest paths for a car-like robot has already an history. The pioneering result has been achieved by Dubins who characterized the shape of the shortest paths for a particle subject to curvature constraints [33]. Starting from this result, Cockayne and Hall have computed the accessibility set for this model of particle [27] (i.e., the domain of the plane reachable by paths with bounded length). The particle model corresponds to a car-like robot moving forward with a constant velocity. More recently, Reeds and Shepp have provided a sufficient family of 48 shortest paths for a car-like system moving both forward and backward [87]: optimal paths are constituted by a finite sequence of at most five straight line segments or arcs of a circle of radius 1.

Then the problem has been revisited from a control theory point of view: Sussmann and Tang [116] and Boissonnat, Cerezo and Leblond [15] independently provided a new proof of Reeds and Shepp's result. In addition Sussmann and Tang reduced the sufficient family to 46 canonical paths.

At the same time, Souères and Laumond, using these results, computed a synthesis of the shortest paths, i.e. a partition of the manifold $\mathbf{R}^2 \times S^1$ into cells reachable by only one type of shortest path (among the 46 ones) [106]. They also computed the exact shape of the shortest path metric, i.e., the shape of the domain in $\mathbf{R}^2 \times S^1$ reachable by paths with bounded length [70].

The projection of that domain on $\mathbf{R}^2$ corresponds to the accessibility domain in the plane; it has been computed by Souères, Fourquet and Laumond [107] who then extended the result of Cockayne and Hall [27] to the car moving also backward; the authors provide also a synthesis of the shortest path from an initial configuration to a point in the plane. On the other hand, with Bui and Boissonnat, they apply the same techniques to compute a synthesis in $\mathbf{R}^2 \times S^1$ for a car moving only forward [20]. Boissonnat and Bui then derive a synthesis of the shortest path to a point in $\mathbf{R}^2$ for this case [21].

Finally, Moutarlier, Mirtich and Canny explored general analytic tools to compute shortest paths to some sub-manifolds of $\mathbf{R}^2 \times S^1$ [78].

In this annex we consider a point robot moving either forward or both forward and backward amidst polygonal obstacles: according to the history above, we will call Dubins car the car-like robot moving only forward ($v = 1$) and Reeds&Shepp car the car-like robot moving both forward and backward ($|v| = 1$). For both systems we propose three geometric algorithms to:

- compute the shortest path from an initial configuration (x_i, y_i, θ_i) to a goal position (x_g, y_g);
- compute the shortest path from a configuration to a segment;
- compute the domains of the plane visible from a start configuration in presence of obstacles.

Note that the results in [107] and [21] lead to efficient algorithms to compute the shortest path to a position. We just propose an alternative geometric way for this computation (Section C.2). Moreover the results presented in [78] allow to design a procedure to compute the shortest path to a line segment for Reeds&Shepp car. Again we propose here a geometric and more efficient algorithm to solve this problem and we extend the results to Dubins car (Section C.3).

Notations The paths are described using Sussmann and Tang notation [116]: capital letters denote either a straight line segment (S) or an arc of a circle of radius 1 (C). The symbol | between two letters indicates the presence of a cusp. To specify the direction of rotation the letter C will be replaced by l for left turn and by r for right turn and the superscript $+$ ($-$) will denote forward (backward) motion[28]. Subscripts represent positive real numbers giving the length of each piece. The sufficient family of

[28] In Dubins car case the superscript will be omitted being it always +

shortest paths can be summarized by the following nine words:

$$\begin{array}{ll}
C_a|C_b|C_e & 0 \leq a < \pi, 0 \leq b < \pi, 0 \leq e < \pi, \\
C_aS_dC_e & 0 \leq a < \frac{\pi}{2}, 0 \leq d, 0 \leq e < \frac{\pi}{2}, \\
C_a|C_bC_e \text{ ou } C_aC_b|C_e & 0 \leq a < b, 0 \leq b < \frac{\pi}{2}, 0 \leq e < b, \\
C_a|C_bC_b|C_e & 0 \leq a < b, 0 \leq b < \frac{\pi}{2}, 0 \leq e < b, \\
C_aC_b|C_bC_e & 0 \leq a < b, 0 \leq b < \frac{\pi}{2}, 0 \leq e < b, \\
C_a|C_{\frac{\pi}{2}}S_dC_{\frac{\pi}{2}}|C_e & 0 \leq a < \frac{\pi}{2}, 0 \leq d, 0 \leq e < \frac{\pi}{2}, \\
C_a|C_{\frac{\pi}{2}}S_dC_e \text{ ou } C_aS_d|C_{\frac{\pi}{2}}C_e & 0 \leq a < \frac{\pi}{2}, 0 \leq d, 0 \leq e < \frac{\pi}{2}.
\end{array} \tag{22}$$

C.2 Shortest paths to a position.

This section recalls the results presented in [107] and [21]: they provide a synthesis of the shortest path to a position for Reeds&Shepp car and Dubins car respectively.

We deduce from these results a geometric construction of the shortest paths which will be used in the next sections.

C.2.1 Background

Reeds&Shepp car In [107] Souères, Fourquet and Laumond proved that only three families of Reeds&Shepp paths may be optimal to reach a position (x, y) from the origin of the configuration space $(0, 0, 0)$. These families induce a partition of the set of positions $\mathbf{R}^2$ in domains reachable by a given family of paths (see Fig. 23).

Moreover from an optimal path $\mathcal{P}$ it is possible to build three other optimal paths isometric to $\mathcal{P}$ and leading to points symmetric to the point reached by $\mathcal{P}$ with respect to the x-axis, the y-axis and the origin of the coordinate axes [107]. For any point not belonging to the y-axis, there is a *unique* Reeds&Shepp shortest path constituted by:

- two arcs of a circle of minimum radius connected by a cusp and followed by a straight line segment (this family is denoted by $C|C_{\frac{\pi}{2}}S$) for points belonging to Region 1. A point in the first quadrant belonging to this region is reached by a path $l_u^- l_{\frac{\pi}{2}}^+ s_d^+$; its coordinates verify:

$$\begin{cases} x(u, d) = \cos u - (2 + d)\sin u \\ y(u, d) = \sin u + (2 + d)\cos u - 1 \end{cases} \tag{23}$$

  with $0 \leq u \leq \arctan \frac{1}{2}$.

- two arcs of a circle of minimum radius connected by a cusp (family $C|C$) for points belonging to Region 2. A point in the first quadrant

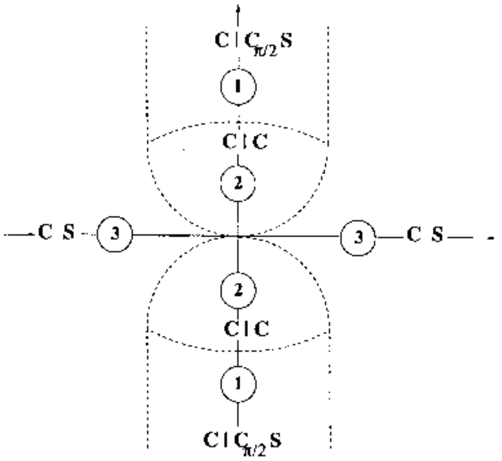


Figure 23: Reeds&Shepp car: Synthesis of the shortest paths to a position

is reached by a path $l_u^- l_v^+$; its coordinates verify:

$$\begin{cases} x(u,v) = -2\sin u + \sin(u+v) \\ y(u,v) = 2\cos u - \cos(u+v) - 1 \end{cases} \tag{24}$$

with $0 \leq u \leq \frac{\pi}{6}$ and $0 \leq v \leq \frac{\pi}{2}$.

- an arc of circle of minimum radius followed by a straight line segment (family CS) for points belonging to Region 3. A point in the first quadrant is reached by a path $l_v^+ s_d^+$; its coordinates verify:

$$\begin{cases} x(v,d) = \sin v + d\cos v \\ y(v,d) = -\cos v + d\sin v + 1 \end{cases} \tag{25}$$

with $0 \leq v \leq \frac{\pi}{2}$.

Due to the symmetry properties, there exist exactly two shortest paths, belonging to the same family, reaching a given position on the y-axis.

Note that the values of the parameters vary continuously from one domain to another, that is the points on the boundaries between two (or three) domains are always reached by a *unique* path that can be considered as belonging simultaneously to the two (three) neighboring domains for appropriate values of the parameters.

Dubins car With the same approach, Boissonnat and Bui [21] provided a similar synthesis for Dubins car. In this case only two families of Dubins paths can be optimal and the x-axis is the axis of symmetry (see Fig. 24); for any point not belonging to the negative half-axis Ox, there is a *unique* Dubins shortest path composed by:

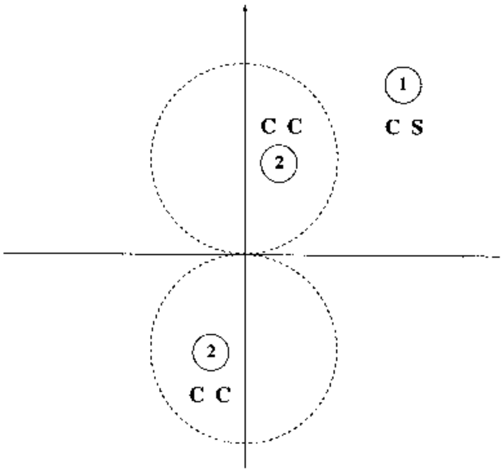


Figure 24: Dubins car: Synthesis of the shortest paths to a position

- an arc of a circle of minimum radius followed by a straight line segment (CS family) for points belonging to Region 1. A point belonging to the upper half-plane is reached by a path $l_v s_d$; its coordinates verify:

$$\begin{cases} x(v,d) = \sin v + d\cos v \\ y(v,d) = 1 - \cos v + d\sin v \end{cases} \tag{26}$$

with $0 \leq v < 2\pi$,

- two arcs of circle of minimum radius (CC family) for points belonging to Region 2. A point belonging to the upper half-plane is reached by a path $r_u l_v$; its coordinates verify:

$$\begin{cases} x(u,v) = 2\sin u - \sin(u-v) \\ y(u,v) = -1 + 2\cos u - \cos(u-v) \end{cases} \tag{27}$$

with $0 \leq u < v$ and $\pi < v < 2\pi$.

For points with coordinates $x < 0$ and $y = 0$ there are two equivalent (and symmetric) shortest paths in the family CS.

C.2.2 Geometric construction

From the results described so far we can derive the following geometric constructions to compute the shortest path to a point P. According to the symmetry properties, P is assumed to be in the first quadrant for Reeds&Shepp car and in the upper half-plane for Dubins car.

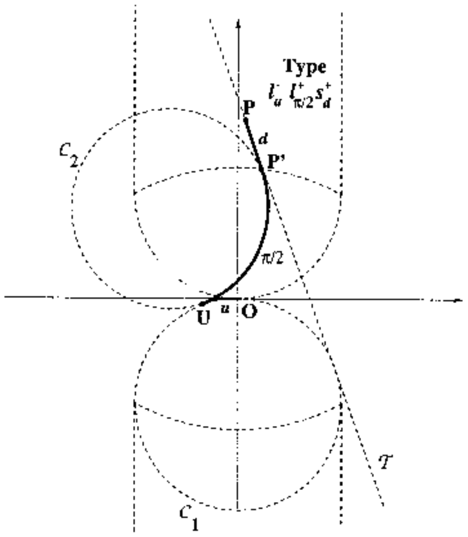


Figure 25: Reeds&Shepp car: The shortest path to a point in Region 1

Reeds&Shepp car

- Point in Region 1. Type of path $l_u^- l_{\pi/2}^+ s_d^+$.
  First the tangent line $\mathcal{T}$ to the circle C_1 (Fig. 25) of radius 1 and center $(0, -1)$, passing through the point $P(x, y)$ is traced, then the center of the circle C_2 of radius 1, tangent to C_1 and to $\mathcal{T}$, is computed. The value of the parameter u is given by the length of the arc $\overset{\frown}{OU}$. Parameter d is determined by the length of the segment $\overline{P'P}$.

- Point in Region 2. Type of path $l_u^- l_v^+$.
  Paths leading to points in Region 2 are computed by finding the center of the unit circle C tangent to the circle C_1 and passing through the point P. As before the value of the parameter u is given by the length of the arc $\overset{\frown}{OU}$, while parameter v is equal to the length of the arc $\overset{\frown}{UP}$(see Fig. 26).

- Point in Region 3. Type of path $l_v^+ s_d^+$.
  Figure 27 shows that paths in Region 3 are obtained by drawing the line $\mathcal{T}$ tangent to the circle C_2 and passing through P. Parameters v and d are given, respectively, by the length of the arc $\overset{\frown}{OV}$ and of the segment $\overline{VP}$.

Note that, in the worst case, only three comparisons between reals are requested to decide to which region the point P belongs.

Dubins car

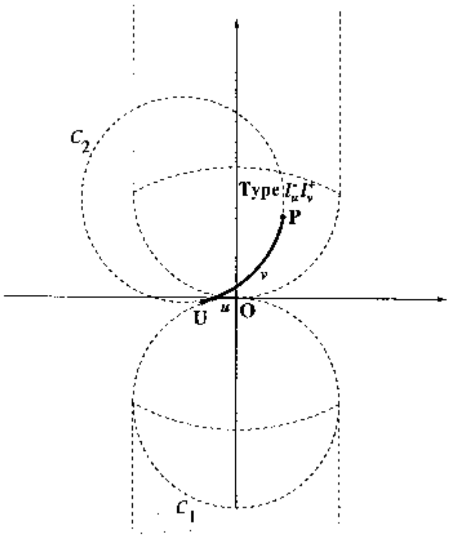


Figure 26: Reeds&Shepp car: The shortest path to a point in Region 2

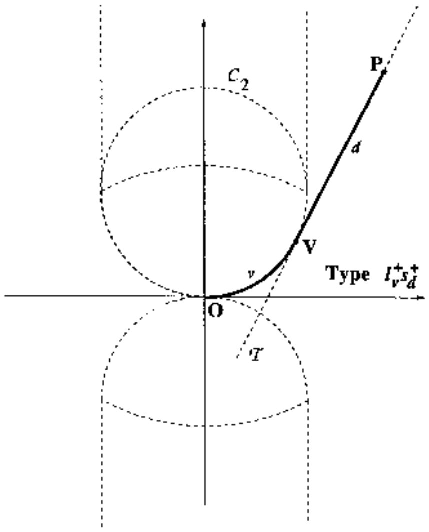


Figure 27: Reeds&Shepp car: The shortest path to a point in Region 3

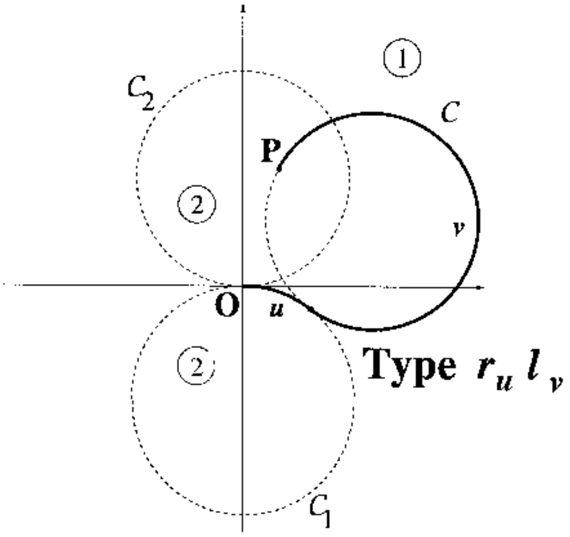


Figure 28: Dubins car: The shortest path to a point in Region 2

- Point in Region 1. Type of path $l_v s_d$.
  The construction is the same as for points belonging to Region 3 of Reeds&Shepp car case.

- Point in Region 2. Type of path $r_u l_v$.
  Paths leading to points in Region 2 are computed by finding the center of the unit circle C tangent to the circle C_1 and passing through the point P (see Figure 28).

Remark: Any sub-path of a shortest path from a configuration to a *configuration* is a shortest path. We should keep in mind that this property does not hold for the shortest paths from a configuration to a *position*.

C.3 Shortest paths to a segment.

Figure 29 shows the sets of positions reachable by paths of length ℓ for different values of ℓ. The corresponding isometric curves (also called "wave fronts") and their geometric structure are basic for the computation of the shortest paths starting from the origin of the configuration space to a given segment. Indeed, the length of the shortest path to a segment is the "radius" ℓ of the first wave front tangent to the segment.

The wave fronts are obtained by expressing the length ℓ of the path in each region as a function of the two parameters characterizing the path and operating an appropriate change of variables. CS and $C|C_{\frac{\pi}{2}}S$ paths generate arcs of an involute of a circle; $C|C$ paths generate arcs of a circle of

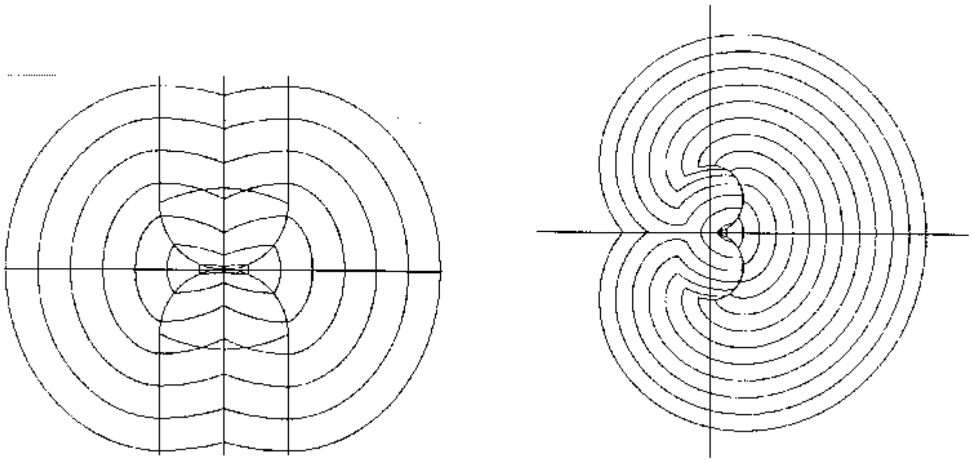


Figure 29: Sets of reachable positions in the plane: Reeds&Shepp car (left) and Dubins car (right)

radius 2; CC paths generate arcs of cardioids. The wave fronts then appear as a sequence of such arcs.

Remark: Notice that the wave fronts for Reeds&Shepp car are always closed curves while they may be simple open and not necessarily connected arcs for Dubins car.

The geometric construction of shortest paths to a segment requires to compute the endpoints of the various wave front arcs and the critical values changing their geometric structure.

C.3.1 Reeds&Shepp car

Due to the symmetry properties, the problem can be first attacked by assuming that the segment lies in the first quadrant. This is particularly convenient from a computational point of view: the slope of the tangent to the wave front in the first quadrant is monotonic (Sec. C.3.1), this allows to decide the region in which the tangency will occur from the slope of the line supporting the segment. The general case will be solved by considering symmetry properties (Section C.3.1).

Shape of the wave-fronts in the first quadrant The set of positions reachable by a path of length ℓ is described by three different curves in the three regions of the first quadrant. Noting that in Region 1, $\ell = u + \frac{\pi}{2} + d$. The change of variable $d = \ell - u - \frac{\pi}{2}$ in the system (23) leads to:

$$\begin{cases} x(u,\ell) = \cos u - (2 + \ell - u - \frac{\pi}{2}) \sin u \\ y(u,\ell) = \sin u + (2 + \ell - u - \frac{\pi}{2}) \cos u - 1 \end{cases} \tag{28}$$

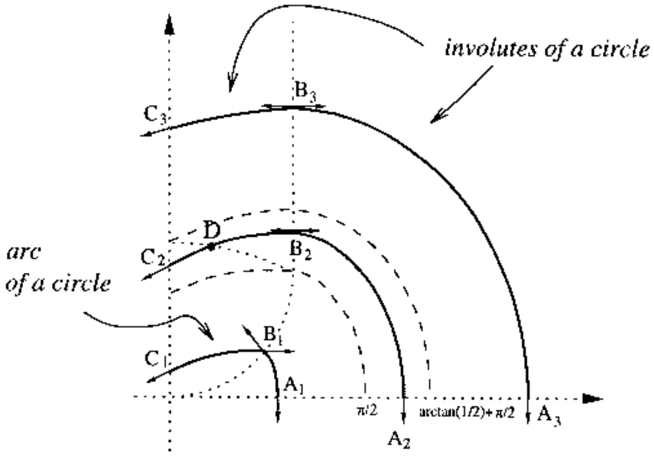


Figure 30: Wave fronts in the first quadrant (Reeds&Shepp car)

For ℓ constant this gives the equation of an involute of a circle.

The length of a path in Region 2 is $\ell = u + v$. Replacing the expression $u + v$ by ℓ in (24) we obtain:

$$\begin{cases} x(u,\ell) = -2\sin u + \sin \ell \\ y(u,\ell) = 2\cos u - \cos \ell - 1 \end{cases} \tag{29}$$

Assuming ℓ constant, the parametric equation of a circle of radius 2 is obtained.

From the expression of the path length $\ell = v + d$ in the third region, the change of variable $d = \ell - v$ in (25) gives:

$$\begin{cases} x(v,\ell) = \sin v + (\ell - v)\cos v \\ y(v,\ell) = -\cos v + (\ell - v)\sin v + 1 \end{cases} \tag{30}$$

The system (30) leads, for ℓ constant, to the equation of an involute of a circle.

Therefore the equation of the wave front in the first quadrant can be easily computed from the systems above. Its shape is shown in Fig. 30. According to the value of ℓ, the following description of the wave front structure can be given:

- $\ell \leq \frac{\pi}{2}$: concatenation of an involute of a circle and an arc of a circle;
- $\frac{\pi}{2} < \ell < \arctan(\frac{1}{2}) + \frac{\pi}{2}$: concatenation of two involutes of a circle and an arc of a circle;
- $\ell \geq \arctan(\frac{1}{2}) + \frac{\pi}{2}$: concatenation of two involutes of a circle.

The following table gives the angle of the half-tangent at the boundary points A_i, B_i, C_i $(i = 1,2,3)$ in Fig. 30:

Case	$\ell \leq \frac{\pi}{2}$			
Points	A_1	B_1^-	B_1^+	C_1
Angle	$-\frac{\pi}{2}$	$v - \frac{\pi}{2}$	0	u

Case	$\frac{\pi}{2} \leq \ell \leq \arctan\frac{1}{2} + \frac{\pi}{2}$		
Points	A_2	B_2	C_2
Angle	$-\frac{\pi}{2}$	0	u

Case	$\ell > \arctan\frac{1}{2} + \frac{\pi}{2}$		
Points	A_3	B_3	C_3
Angle	$-\frac{\pi}{2}$	0	u

In all cases the slope of the tangent line is monotonic in the first quadrant. Moreover, at point D in Fig. 30 (connection of an involute of a circle and an arc of a circle), the first derivative is still continuous. Therefore looking at the slope of the line supporting the segment it is easy to decide in which region the tangency will occur.

Geometric construction We then solve the problem for each region:

- Regions 1 and 3

  The isodistance curves in regions 1 and 3 are, respectively, the involutes of the circles C_1 and C_2 (see Fig. 31) and have the nice property that the line $\mathcal{T}$ tangent to the circles and passing through the point T is perpendicular to the line tangent to the involute at the same point [48]. The segments of length d constituting the last part of the paths leading to regions 1 and 3 belongs to the line $\mathcal{T}$, as shown in the geometric construction of the shortest paths to points in these regions. The algorithm to find the shortest paths to the tangency points becomes then very easy: in Region 1 the parameter u is such that the robot final orientation is perpendicular to the tangent line supporting the segment; therefore u is such that $\tan(u) = m$, where m is the slope of the tangent[29]. Parameter d is equal to the length of the segment $\overline{T'T}$ (Fig. 31, top). Analogously to the previous case, the parameter v in Region 3, is defined by $\tan(v) = -\frac{1}{m}$, where m has the same meaning as above[30]. Looking at Fig. 31 (bottom) it is easy to conclude that the length of the segment $\overline{VT}$ determines the value of the parameter d.

- Region 2

  In Region 2 the wave front is an arc of the circle C_T of radius 2 the

[29] This means that the car starts with a backward motion until it becomes parallel to the segment.

[30] In this case the robot final orientation is perpendicular to the segment.

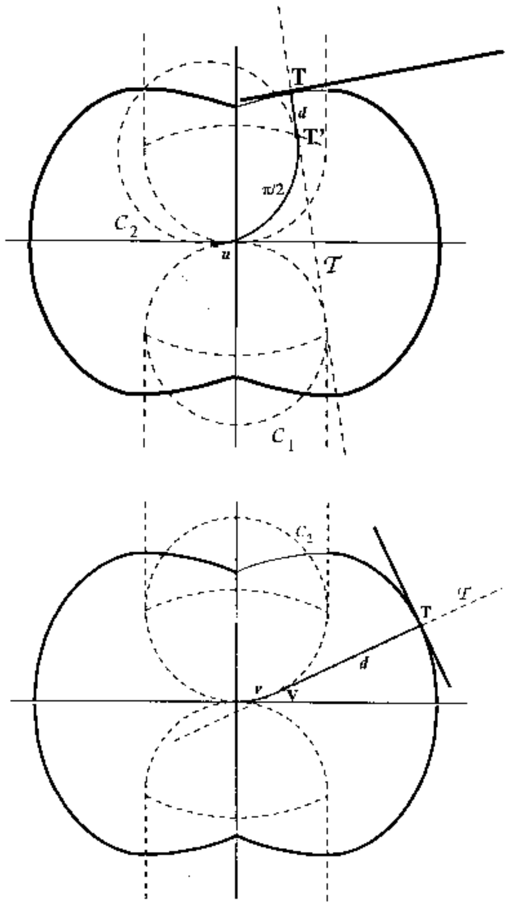


Figure 31: Reeds&Shepp car: Tangency in Region 1 (top) or in Region 3 (bottom)

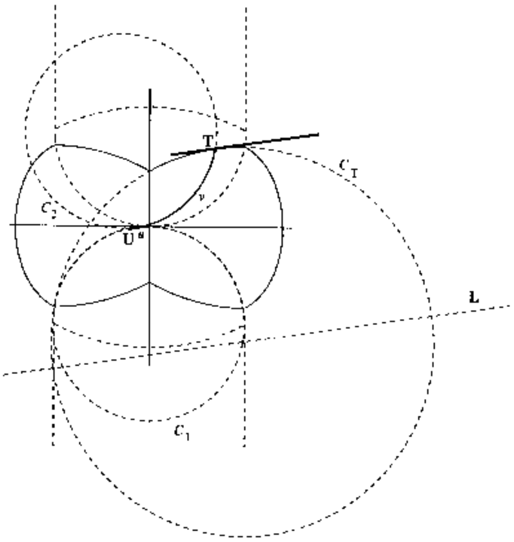


Figure 32: Reeds&Shepp car: Tangency point in Region 2

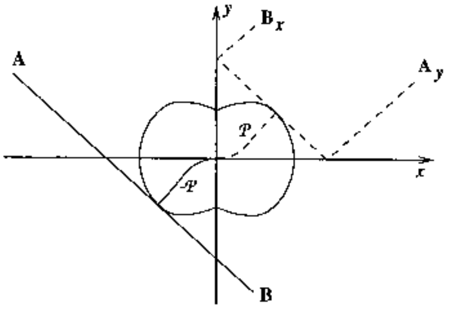


Figure 33: Shortest path computation by symmetry

center of which is on the unit circle $\mathcal{C}_1$ and on the line L parallel to the tangent line supporting the segment and located at a distance 2 from it, as shown in Fig. 32. Once found the center of this circle the tangency point is computed. Anyway, looking at the geometric construction shown in the figure it is possible to prove that the robot orientation at point U is parallel to the tangent line supporting the segment. The center of the circle $\mathcal{C}_2$ can be then determined using this fact and the computation of the tangent point becomes trivial.

Symmetry property We have solved the problem of finding the shortest path to a segment lying in the first quadrant.

The problem of finding the shortest path to a segment in the plane is solved by computing the symmetric image in the first quadrant of the segment. Then the shortest path to this polygonal image is computed. The found path is then transformed by the appropriate symmetry.

Figure 33 shows the case of a segment traversing the second, third and forth quadrant. The dashed polygonal line denotes its image in the first quadrant and the path $\mathcal{P}$ is the shortest path reaching the line. Since the path $\mathcal{P}$ reaches the line at a point belonging to the image of that part of the segment lying in the third quadrant, the shortest path to the segment is the image of $\mathcal{P}$ obtained by symmetry with respect to the origin.

The general rule can be informally summarized: the shortest path to an object (point, segment) in the plane, the image of which in the first quadrant has been computed by a symmetry, is obtained by applying the same symmetry to the shortest path to its image in the first quadrant. The same procedure will be used to find the shadow of a segment in the plane.

C.3.2 Dubins car

By virtue of the symmetry property, we can first assume that the segment lies in the upper half-plane. The geometric construction of the shortest path to a segment is simpler than for Reeds&Shepp car. Indeed the case of Region 1 of Dubins car is exactly the same as the case of Region 3 of Reeds&Shepp car; the only difference is that the former covers the later. As we will show, the remaining case gives rise to an easy construction.

Shape of the wave fronts Some of the following computations appear in [27, 21]. Other ones are original; they are developed in the proof of Lemma below. All of them can be easily derived from the equations (26) and (27) with the help of any computer algebra system.

Region 1: For $\ell \leq \frac{3\pi}{2} + 1$, the wave front is an arc of involute of a circle starting on the point $(\ell, 0)$ on the x-axis and ending on $\mathcal{C}_2$; at $\ell = \frac{3\pi}{2} + 1$ the involute is tangent to the x-axis. For $\frac{3\pi}{2} + 1 < \ell < 2\pi$, the wave front keeps only the two arcs of the involute lying in the upper half-plane. For $\ell > 2\pi$ it remains just a piece of the involute (see Fig. 34).

Region 2: The paths reaching points in Region 2 belong to the family $r_u l_v$. Their length ℓ is $u + v$ with $0 \leq u < v$ and $\pi < v < 2\pi$. Replacing $u - v$ by $2u - \ell$ in equations (27) we obtain:

$$\begin{cases} x(u, \ell) = 2\sin u - \sin(2u - \ell) \\ y(u, \ell) = -1 + 2\cos u - \cos(2u - \ell) \end{cases} \tag{31}$$

These are the equations of a cardioid. Due to the inequalities on u and v, a wave front containing such an arc should have its radius ℓ strictly greater than π. The maximum value ℓ_c is computed in the lemma below.

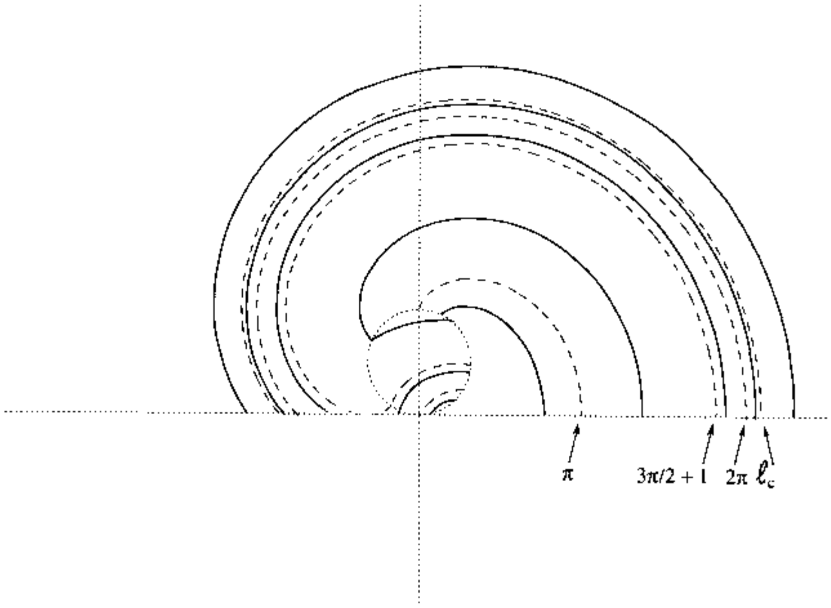


Figure 34: Wave-fronts in the upper half-plane (Dubins car)

The following table summarizes the critical values of ℓ that change the geometric structure of the wave fronts (Figure 34) in the upper half-plane:

Case	$\ell \leq \pi$	$\pi < \ell < \frac{3}{2}\pi + 1$	$\frac{3}{2}\pi + 1 < \ell < 2\pi$	$2\pi < \ell < \ell_c$	$\ell_c < \ell$
# Arcs of involute in Region 1	1	1	2	1	1
# Arcs of cardioid in Region 2	0	1	1	1	0

In this region, the geometric construction of the shortest path to a segment is much more easy than in the cases previously addressed. Nevertheless, its proof requires a technical lemma.

Geometric construction of the shortest path to a segment The geometric construction of the shortest path to a segment lying in Region 1 (Figure 35) follows exactly the principle described for Region 3 of Reeds&Shepp car.

The case of segments lying in Region 2 is addressed in the following lemma.

Lemma 1 *The shortest path to a segment S lying in Region 2 for Dubins car is one of the two shortest paths reaching its endpoints.*

Proof 4 *Let us consider the upper half-plane. Region 2 for Dubins car in this domain is the disk of radius 1 centered at $(0,1)$. Let C_2 be the associated*

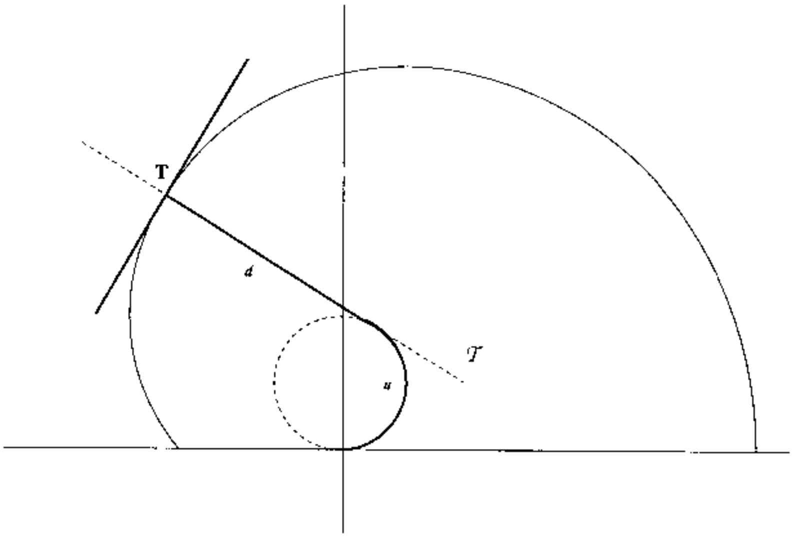


Figure 35: Dubins car: Tangency in Region 1

circle. A point of coordinates (x, y) of the disk is reached from the origin $(0, 0, 0)$ by Dubins car along the shortest path $r_u l_v$ such that u and v verifies the system (27):

$$\left\{ \begin{array}{l} x(u, v) = 2\sin u - \sin(u - v) \\ y(u, v) = -1 + 2\cos u - \cos(u - v) \end{array} \right.$$

with $0 \leq u < v$ and $\pi < v < 2\pi$. Moreover u should be lesser than $\frac{\pi}{3}$.

For a fixed value of u the system corresponds to the equation of a circle; the endpoints on C_2 of the corresponding arc of a circle is obtained for $v_1 = \frac{3\pi}{2} + \frac{u}{2} - \arccos(2\sin\frac{u}{2})$ and $v_2 = \frac{3\pi}{2} + \frac{u}{2} + \arccos(2\sin\frac{u}{2})$. The maximum length of the path (for u fixed) is then $u + v_2$. By derivating the later expression with respect to u, we compute the value of u maximizing the length of all the paths reaching a point in Region 2; we may then derive the radius of the latest wave front covering Region 2; this radius is: $\ell_c = \frac{3}{2}\pi + 3\arccos\frac{3\sqrt{6}}{8} + \arccos\frac{\sqrt{10}}{4}$.

Then Region 2 is covered by the wave fronts whose radii range from π to ℓ_c. From now, the principle of the proof is to show that the cardioid arcs span Region 2 along a "concave" foliation[31] (see Figure 36, left).

[31] A *foliation* of a n-dimensional manifold $\mathcal{M}$ is a family $\mathcal{L}_\alpha, \alpha \in I$ of arcwise connected q-dimensional sub-manifolds ($q < n$), called *leaves*, of $\mathcal{M}$ such that:

- $\mathcal{L}_\alpha \cap \mathcal{L}_{\alpha'} = \emptyset$ if $\alpha \neq \alpha'$
- $\cup_{\alpha \in I} \mathcal{L}_\alpha = \mathcal{M}$
- every point in $\mathcal{M}$ has a local coordinate system such that $n - q$ coordinates are constant.

Let $\ell = u + v$ be the length of the path; ℓ varies from $\frac{3}{2}(\pi + u) - \arccos(2\sin\frac{u}{2})$ to $\frac{3}{2}(\pi + u) + \arccos(2\sin\frac{u}{2})$.

Replacing $u - v$ by $2u - \ell$ in equations (27) we obtain the system (31) describing a cardioid. The endpoints of the arc of cardioid on the circle $\mathcal{C}_2$ are given for $u = 0$ and some critical value $u_c(\ell)$[32]*. Let us denote the arc of cardioid by $\mathcal{L}_\ell$. According to the unicity of the shortest Dubins path $r_u l_v$ in Region 2, there is only one $\mathcal{L}_\ell$ passing through a given point in Region 2. All the $\mathcal{L}_\ell$ clearly span Region 2. Therefore each point in Region 2 can be located by $\ell \in]\pi, \ell_c[$ and a real number $u \in]0, u_c(\ell)[$. In this coordinate system all the points in $\mathcal{L}_\ell$ have a constant coordinate. The set of all the $\mathcal{L}_\ell$ constitutes a foliation of Region 2.*

By eliminating u in the system (31) we get the equation of the wave front of radius ℓ:

$$f(x,y) = -x\sin\ell - (y+1)\cos\ell + 3 - \frac{1}{2}x^2 - \frac{1}{2}y^2 - y - 2(1 - \frac{1}{4}x^2 - \frac{1}{4}y^2 - \frac{1}{4}y^2)^2$$

For a given ℓ, each arc of cardioid $\mathcal{L}_\ell$ splits Region 2 into two sub-domains: the sub-domain $\mathcal{R}_\ell^-$ of points reachable by shortest paths of length strictly lesser than ℓ and the sub-domain $\mathcal{R}_\ell^+$ of points reachable by shortest paths of length strictly greater than ℓ. Let us compute the value of f at the point of coordinates $(0, 2)$: $f(0, 2) = -3(\cos\ell + 1)$ which is always negative. Then $\mathcal{R}_\ell^-$ always contains the point of coordinates $(0, 2)$: $\mathcal{R}_\ell^-$ lies above the wave front.

Therefore, let us consider a straight line segment secant to an arc $\mathcal{L}_\ell$; one of its endpoints lies necessarily in $\mathcal{R}_\ell^-$. This endpoint is the same for any $\mathcal{L}_\ell$ intersecting the segment. Then, if the segment is not tangent to some leave $\mathcal{L}_\ell$, this endpoint of the segment is the endpoint of the shortest path reaching the segment.

It just remains to consider the tangency case (Figure 36, right). Due to the geometric definition of a cardioid, the concavity of all the $\mathcal{L}_\ell$ at any point of coordinates (ℓ, u) (in the coordinate system of the foliation) is directed towards the endpoint of the path r_u located on $\mathcal{C}_1$ (Figure 28). This endpoint is below $\mathcal{L}_\ell$. This means that the line tangent to $\mathcal{L}_\ell$ at any point lies above $\mathcal{L}_\ell$. Finally any straight line segment included in Region 2 and tangent to some $\mathcal{L}_\ell$ is included in $\mathcal{R}_\ell^-$. Then the abscissa of the endpoints of the segment (in the coordinate system of the foliation) is lesser than ℓ. The first wave front

[32] For takers, the value of $u_c(\ell)$ is

$$2\arctan\left(\%1^{1/3} + \frac{4\cos(l) - 1}{3(2+\cos(l))^2\,\%1^{1/3}} + \frac{\sin(l)}{2+\cos(l)}\right)$$

$$\%1 := -3\,\frac{\sin(l)}{(2+\cos(l))^3} + \frac{1}{9}\,\frac{\sqrt{183 - 192\cos(l)}}{(2+\cos(l))^2} \quad \text{(Thank you Maple...)}$$

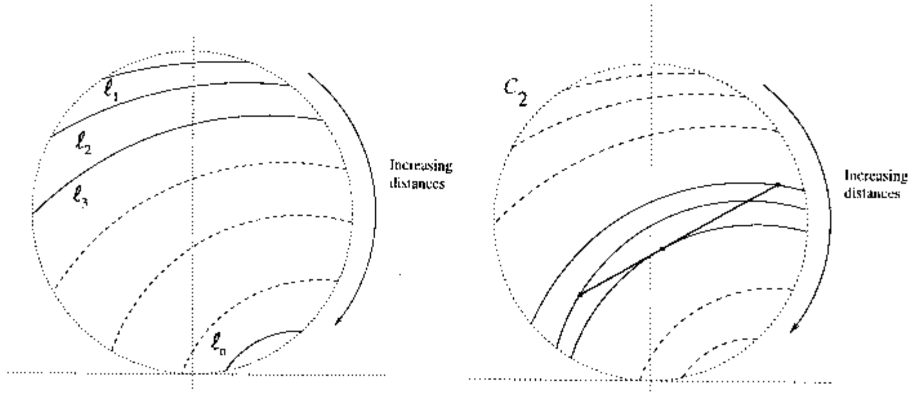


Figure 36: Foliation of Region 2 for Dubins car ($\ell_1 < \ell_2 < \ell_3 < ... < \ell_n$)

reaching the segment meets the latter necessarily at the endpoint with the lowest abscissa ℓ.

This lemma gives an easy way to compute the shortest path to a segment lying in Region 2: it suffices to compute the shortest paths to its endpoints and to choose the shortest one.

Finally, to compute the shortest path to a segment for Dubins car, it suffices to compute its symmetric image in the upper half-plane; each segment of the image (almost two pieces) is splitted into sub-segments lying either in Region 1 or in Region 2. The shortest paths is then computed for each of these sub-segments. It remains to choose the shortest one among them.

C.4 Examples

Figures 37 and 38 shows various examples of shortest paths to a straight line segment for both Reeds&Shepp carand Dubins car. Such obstacle distances are the core of procedures allowing real-time obstacle avoidance and path smoothing in motion planning for mobile robots [53, 49].

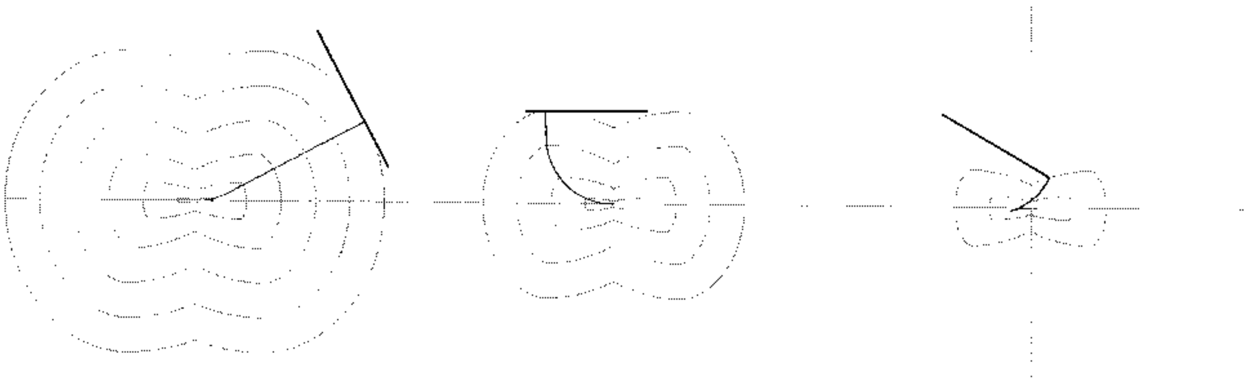


Figure 37: Examples of shortest paths to segments for the Reeds&Shepp car

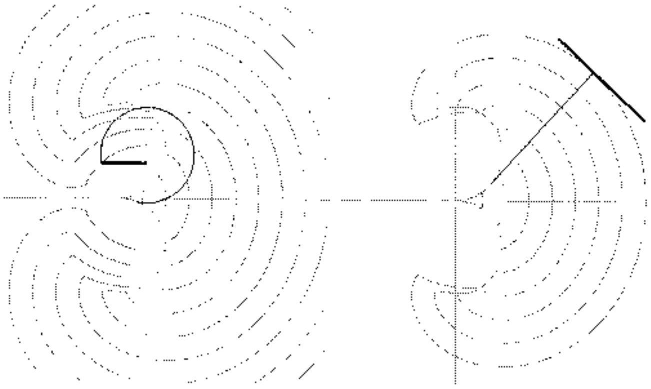


Figure 38: Examples of shortest paths to segments for the Dubins car

D On the topological property of Steer_{sin}

In Section 5.2.2 we have seen that Steer_{sin} is an exact method based on sinusoidal controls allowing to steer a chained form system. However, we do not dispose of a unique expression of Steer_{sin} verifying the topological property. In this annex we show that it is possible to switch between different $\mathsf{Steer}_{sin}^{a_1}$ to integrate such a steering method within the general nonholonomic path planning schemes introduced in Sections 5.3 and 5.4.

Let us consider the two input chained form system (8) introduced in Section 4.3:

$$\begin{cases} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2.v_1 \\ \vdots &= \vdots \\ \dot{z}_n &= z_{n-1}.v_1 \end{cases}$$

$\mathsf{Steer}_{sin}^{a_1}$ is defined by:

$$\begin{cases} v_1(t) &= a_0 + a_1 \sin \omega t \\ v_2(t) &= b_0 + b_1 \cos \omega t + b_2 \cos 2\omega t + \ldots b_{n-2} \cos(n-2)\omega t \end{cases}$$

We have proved that for a given a_1 small enough, the maximal gap between Z^{start} and the path $\mathsf{Steer}_{sin}^{a_1}(Z^{start}, Z^{goal})$ decreases when Z^{goal} tends to Z^{start}. But this gap do not tends to zero. In other words, for a fixed value of a_1, trying to reach closer configurations on the geometric path decreases the risk of collision but does not eliminate it. Moreover to tend this gap to zero we have also to decrease $|a_1|$. But these two decreasings are not independent. Indeed, by changing the value of a_1 we change the steering method $\mathsf{Steer}_{sin}^{a_1}$ and so we change the family of the paths. For a given couple of extremal configurations, a decreasing of a_1 increases in most of the cases the extremal gap between the start point and the path. In other words, in order to reduce the risk of collision we have to choose close goal configurations but we also have to reduce a_1. Which in turn increase again the clearance between the path and the start point. So we have again to bring the goal closer... If the decreasing of $|a_1|$ is too fast with respect to the one of the distance between the start configuration and the current goal, the approximation algorithm will not converge.

A strategy for tuning these two decreasings can be integrated in the approximation algorithm (Section 5.3) while respecting its completeness. The following approach has been implemented; it is described with details in [100, 99]. It is based on a lemma giving an account of the distance between a path generated by $\mathsf{Steer}_{sin}^{a_1}$ and its starting point Z^0. Let us denote $z_i(t) - z_i^0$ by $\delta_i(t)$.

Lemma 2 *For any path computed by* $\mathsf{Steer}^{a_1}_{sin}$*, for any* $t \in [0, T]$ *:*

$$\begin{array}{lcl} |\delta_1(t)| & \leq & |a_0T| + |a_1T| = \Delta_1 \\ |\delta_2(t)| & \leq & \sum |b_iT| = \Delta_2 \\ |\delta_{k+1}(t)| & \leq & |z_k^0|\Delta_1 + \ldots + |z_3^0|\Delta_1^{k-2} + (|z_2^0| + \Delta_2)\Delta_1^{k-1} \quad \mathit{withk} > 2 \end{array} \tag{32}$$

Proof 5 *By definition* $\dot{\delta}_1(t) = a_0 + a_1 \sin \omega t$*. Then:*

$$|\delta_1(t)| \leq \int_0^t |\dot{\delta}_1(\tau)| d\tau \leq \int_0^t (|a_0| + |a_1|) \ d\tau \leq |a_0T| + |a_1T|$$

By setting $\Delta_1 = |a_0T| + |a_1T|$ *we have the intermediate result that for all* t*,* $\int_0^t |\dot{\delta}_1(\tau)| d\tau \leq \Delta_1$*. The same reasoning holds to prove that* $|\delta_2(t)| \leq \sum |b_iT|$*.*

Now, for any $k > 2$*:*

$$\delta_{k+1}(t) = \int_0^t z_k(\tau)\dot{z}_1(\tau) \ d\tau = \int_0^t \delta_k(\tau)\dot{z}_1(\tau) \ d\tau + z_k^0 \int_0^t \dot{z}_1(\tau) \ d\tau$$

An upper bound Δ_k *on* $|\delta_k(t)|$ *being given, we get:*

$$|\delta_{k+1}(t)| \leq \Delta_k \int_0^t |\dot{z}_1(\tau)| \ d\tau + |z_k^0| \int_0^t |\dot{z}_1(\tau)| \ d\tau \leq (\Delta_k + |z_k^0|)\Delta_1$$

Then

$$\Delta_{k+1} \leq (\Delta_k + |z_k^0|)\Delta_1$$

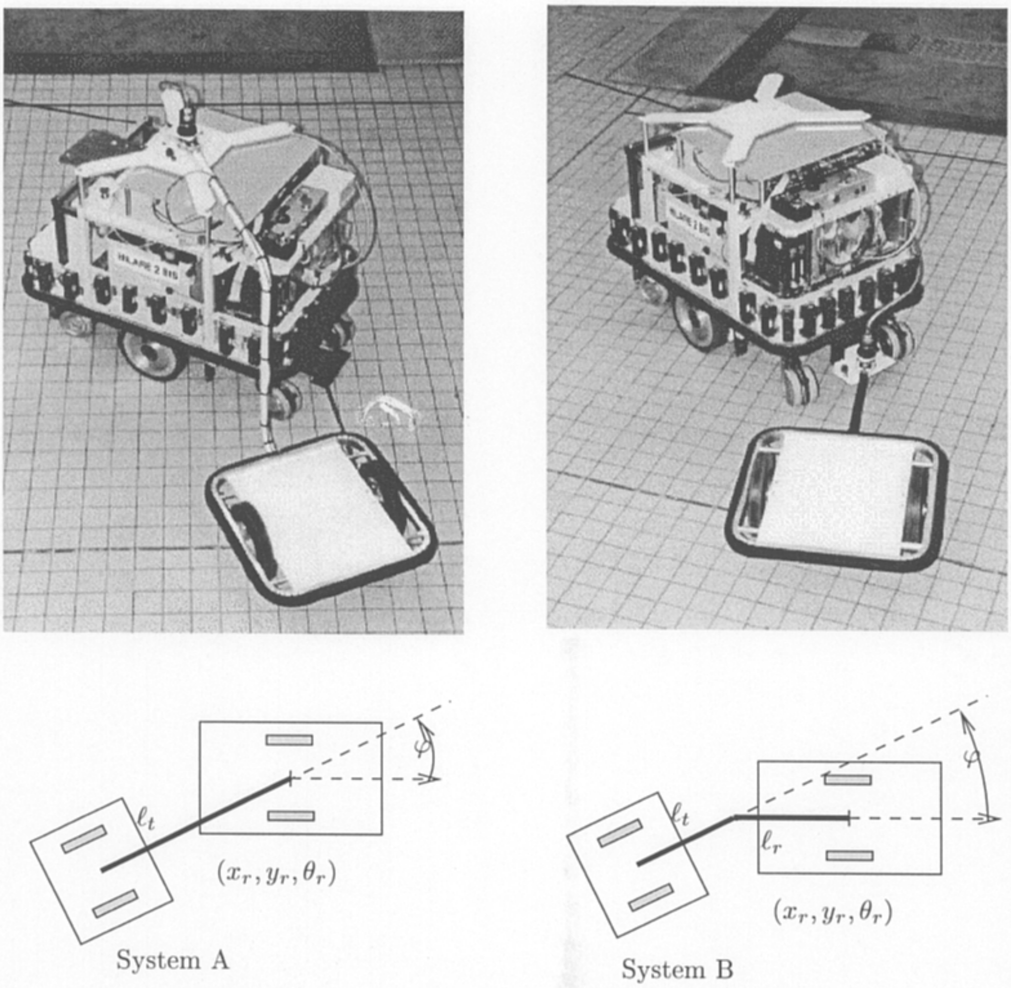
And by recurrence:

$$|\delta_{k+1}(t)| \leq |z_k^0|\Delta_1 + \ldots + |z_3^0|\Delta_1^{k-2} + (|z_2^0| + \Delta_2)\Delta_1^{k-1}$$

Given a start configuration Z^{start}, we first fix the value of a_1 and two other parameters Δ_1^{min} and Δ_2^{min} to some arbitrary values (see [99] for details on initialization). Then we choose a goal configuration on the straight line segment $[Z^{start}, Z^{goal}]$ (or on any collision-free path linking Z^{start} and Z^{goal}]) closer and closer to Z^{start}. This operation decreases the parameters $a_0, b_0, \ldots, b_n$ so it decreases Δ_1 and Δ_2 (the detailed proof of this statement appears in [99, 102]). We continue to bring the goal closer to the initial configuration until a collision-free path is found or until $\Delta_1 \leq \Delta_1^{min}$ and $\Delta_2 \leq \Delta_2^{min}$. In the second case, we substitute a_1, Δ_1^{min} and Δ_2^{min} respectively by $k.a_1$, $k.\Delta_1^{min}$ and $k.\Delta_2^{min}$, with $k < 1$ and we start the above operations again. The new starting path may or may not go further away from Z^{start} than the previous one but in any case, from equations (32) we have the guarantee that following this strategy, the computed path will lie closer and closer to Z^{start}. We have then the guarantee of finding a collision-free path.

E From paths to motions: practical issues on Hilare pulling a trailer

In this annex we present complementary issues dealing with motion execution. Up to now we have presented techniques for nonholonomic *path* planning. Once a path had been planned, it has to be transformed into a *trajectory*. Then the trajectory has to be executed via a *feedback control*. Here we present solutions which have been developed for the experimental platform Hilare-2-bis of LAAS. Hilare-2-bis is a two driving-wheel mobile robot belonging to the family of mobile robot Hilare growing at LAAS since 1976 [41].



System A

System B

Figure 39: Hilare with its trailer

E.1 Hilare and its trailer

In the following experimentation we use proprioceptive sensors: the odometer (based on optical encoders on dedicated wheels) gives the position (x_r, y_r) and the direction θ_r of the robot w.r.t. a starting configuration; an angular encoder gives the relative direction φ of the trailer w.r.t. the direction of the robot. Three cameras on the ceiling give the initial configuration of the system w.r.t. an absolute frame.

We dispose of two different hooking-up systems A and B (Figure 39). The corresponding control systems are given by:

System A:

$$\left\{\begin{array}{lcl} \dot{x}_r & = & v_r \cos\theta_r \\ \dot{y}_r & = & v_r \sin\theta_r \\ \dot{\theta}_r & = & \omega_r \\ \dot{\varphi} & = & -\frac{v_r}{l_t}\sin\varphi - \omega_r \end{array}\right.$$

System B:

$$\left\{\begin{array}{lcl} \dot{x}_r & = & v_r \cos\theta_r \\ \dot{y}_r & = & v_r \sin\theta_r \\ \dot{\theta}_r & = & \omega_r \\ \dot{\varphi} & = & -\frac{v_r}{l_t}\sin\varphi - \frac{l_r\omega_r}{l_t}\cos\varphi - \omega_r \end{array}\right.$$

where the inputs v_r and ω_r are the linear and angular velocities of the robot. They are submitted to the following constraints: $|v_r| \leq v_{max}$, $|\omega_r| \leq \omega_{max}$, $|\dot{v_r}| \leq \dot{v}_{max}$ and $|\dot{\omega_r}| \leq \dot{\omega}_{max}$. These constraints and the weight of the robot ensures the absence of lateral slipping of the wheels. l_r and l_t are constants defining the geometry of the hooking up system.

Let us notice that although system A is a particular case of system B with $l_r = 0$, both systems have different properties from a control point of view and they have to be studied separately.

The hardware architecture of our experiments (Figure 40) is composed of a Unix workstation and on-board processors, communicating via radio Ethernet. The software architecture is organised in three modules and an interface to control the execution during experiments. The module **TRPLANNER** on the workstation computes a collision-free feasible path according to the approximation scheme described in Section 5.3 and using the steering method Steer_{flat} described in Section 5.2.3. Then the path is sent to the module TRPILO. This latter module first computes a time parameterization on-board, and then samples the corresponding open-loop inputs $(v_r(t), \omega_r(t))$ on a segment of shared memory called *poster*. The module **TRLOCO** reads these data on the poster and computes in real-time the closed loop control of each motor. The position of the initial and final configurations are measured by an absolute localization system composed of three cameras mounted on the ceiling of the experimentation.

E.2 From Path to Trajectory

Once a path between two configurations has been produced, it has to be parameterized by time to take into account the bounds on the velocities and

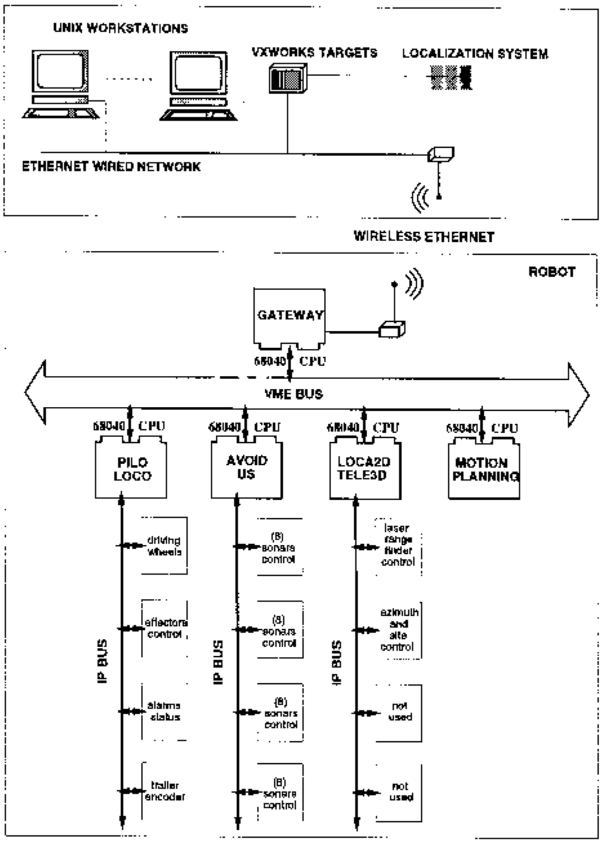


Figure 40: Computer architecture of Hilare-2-bis

accelerations of the robot. This task is performed on board by the module TRPILO. The input of this module is a path $\{(x_r(s), y_r(s), \theta_r(s), \varphi(s)), s \in [s_{start}, s_{end}]\}$, $vmax$, $\dot{v}_{max}$, ω_{max}, and $\dot{omega}_{max}$. The output is an admissible trajectory $((x_r(s(t)), y_r(s(t)), \theta_r(s(t)), \varphi(s(t))), t \in [t_{start}, t_{end}])$, where $s(t)$ is the time parameterization to be computed.

E.2.1 Related work and motivation

Integrating constraints on velocities and accelerations can be done at the planning level. This is the so-called kinodynamic motion planning problem[24, 88]. The methods are based on a discretization of the configuration space and require a perfect knowledge of the $\mathcal{C}$-obstacles. We did not explore this direction because of the computational cost of a search in the phase space of our system which is 6-dimensional.

Transforming a path into a trajectory is a classical problem in robotics. The minimal time parameterization of a given path has been mainly addressed for manipulators (see [90] for an overview). Applications to mobile robots appears in [103]: the computation of a time-optimal motion along a path is used to evaluate the cost of a path through the more ambitious objective to compute optimal trajectories for a mobile robot moving on a terrain. The problem is well understood; formal solutions exist. However the parameterization functions are most of the time described as solutions of differential equations. Their effective computation requires numerical integration. We did it. We got only fragile software. Moreover we can check that the literature does not contain any experimental applications addressing the practical issues of the implementation. At this level, the contribution below is more practical than formal.

The structure of the algorithm we propose here is based on the method described in [104]. However, large variations of the functions $\theta_r(s)$ and $\varphi(s)$ that may be returned by the path planner raise computational issues. Indeed, computing integral curves corresponding to maximal or minimal acceleration requires a small time-step and thus a lot of memory to store the discretized solutions. An inappropriate time discretization combined with big variations of the functions expressing the differential equation may result in significant violations of acceleration bounds. To avoid these troubles and spare memory space, we choose to relax the time optimal condition and to represent acceleration curves over successive intervals of varying size.

E.2.2 Constraints in the phase plane $(s, \dot{s})$

Without loss of generality we consider now the case of a forward motion. By setting $d_v(s) = \sqrt{\frac{dx_r}{ds}(s)^2 + \frac{dy_r}{ds}(s)^2}$ and $d_\omega(s) = \frac{d\theta_r}{ds}(s)$, the velocities and acceleration have the following expressions:

$$\begin{aligned} v &= d_v(s)\dot{s} && (33)\\ \omega &= d_\omega(s)\dot{s} && (34)\\ \dot{v} &= d_v(s)\ddot{s} + \delta_v(s)\dot{s}^2 && (35)\\ \dot{\omega} &= d_\omega(s)\ddot{s} + \delta_\omega(s)\dot{s}^2 && (36) \end{aligned}$$

where

$$\delta_v(s) = \frac{d}{ds}d_v(s) \text{ and } \delta_\omega = \frac{d}{ds}d_\omega(s).$$

Velocity Constraints. The velocity constraints $0 \leq v \leq v_{max}$ and $|\dot{\omega}| \leq \omega_{max}$ are represented by a forbidden area in the *phase plane* $(s, \dot{s})$.

$$\dot{s} \leq \mathrm{Inf}\{\frac{v_{max}}{d_v(s)}, \frac{\omega_{max}}{|d_\omega(s)|}\}$$

We call *velocity saturation curve* the curve obtained when the previous inequality is an equality.

Acceleration constraints. From (35) and (36), $(s, \dot{s})$ being given in the phase plane, the acceleration constraints $|\dot{v}| \leq \dot{v}_{max}$ and $|\dot{\omega}| \leq \dot{\omega}_{max}$ impose $\ddot{s}$ to belong to the intersection $[\alpha(s, \dot{s}), \beta(s, \dot{s})]$ of two intervals. An equivalent condition for this interval not to be empty is (after computations):

$$\dot{s}^2 \leq \frac{\dot{\omega}_{max}|d_v(s)| + \dot{v}_{max}|d_\omega(s)|}{\Delta} \tag{37}$$

with

$$\Delta = d_v(s)\delta_\omega(s) - d_\omega(s)\delta_v(s)$$

The curve corresponding to equality in (37) is called the *maximal velocity curve*. It is denoted by $g(s)$. The signification of this curve can be interpreted as follows. At any point of the path, if the velocity is too high, both acceleration constraints cannot be satisfied. An example of this fact is the case of a car following a road composed of a straight line and a turn of increasing curvature. This situation corresponds to a coefficient δ_ω starting from 0 and increasing along the turn. If the speed of the car is too high, even by braking as much as possible, the angular acceleration cannot be made smaller than its maximal allowed value. This example illustrates the fact that finding a correct parameterization of a path is a global problem that requires knowledge of the path in the future.

E.2.3 Algorithm

Before explaining our algorithm, we need to define the notion of characteristic point, which is a key point of the method. We define then what we call acceleration and deceleration curves.

Characteristic points. From the previous definition, the interval $[\alpha(s, \dot{s}), \beta(s, \dot{s})]$ is empty iff $\dot{s} > g(s)$. Moreover, $\alpha(s, g(s)) = \beta(s, g(s))$ if $d_\omega(s) \neq 0$. In our case, d_v never vanishes, it is a property of our local planner.

We define

$$\kappa(s) = \frac{d\dot{s}}{ds} - \frac{dg}{ds}$$

the difference between the slope of the phase plane trajectory at the maximal velocity curve (i.e. $\frac{d\dot{s}}{ds} = \frac{\alpha(s, g(s))}{g(s)}$) and the slope of the maximal velocity curve.

We say that $(s, \dot{s})$ is an *out-point* if $\kappa(s) > 0$ and an *in-point* if $\kappa(s) < 0$. With these notations, *characteristic points* are defined as points $(s, \dot{s})$ where $\kappa(s^-) > 0$ and $\kappa(s^+) < 0$. At these points and only at these points, a phase plane trajectory can meet the maximum velocity curve without violating the acceleration constraints.

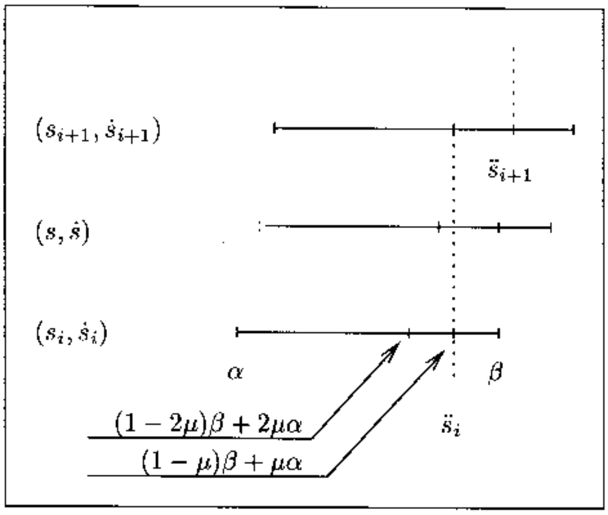

$(s_{i+1}, \dot{s}_{i+1})$
$\ddot{s}_{i+1}$
$(s, \dot{s})$
$(s_i, \dot{s}_i)$
α
β
$(1-2\mu)\beta + 2\mu\alpha$
$\ddot{s}_i$
$(1-\mu)\beta + \mu\alpha$

Figure 41: *Acceleration curve: $\ddot{s}_i$ must remain in the upper interval between s_i and s_{i+1}.*

Acceleration and deceleration curves: From now on we call $\dot{s}$ and $\ddot{s}$ pseudo-velocity and pseudo-acceleration. Let $\mu < 1/4$ a positive real number be given. The key idea of an acceleration (resp. deceleration) curve is to define contiguous intervals of the s-axis where the pseudo-acceleration can be kept constant and in the upper (resp. lower) 2μ-portion of the interval $[\alpha(s,\dot{s}), \beta(s,\dot{s})]$. The size of the intervals is thus automatically adapted to the variation of the coefficients and our representation is far less memory consuming than discretizing integral curves.

The $\ddot{s_0}$-*constant pseudo-acceleration curve* passing by $(s_0, \dot{s_0})$ is represented in the phase plane by a parabola:

$$\dot{s} = \Gamma_{(s_0,\dot{s_0},\ddot{s_0})}(s) = \sqrt{\dot{s_0}^2 + 2\ddot{s_0}(s - s_0)}$$

The acceleration curve starting from a point in the phase plane $(s_0, \dot{s_0})$ is defined by the following algorithm. Let $\ddot{s_i} = (1-\mu)\beta(s_i, \dot{s_i}) + \mu\alpha(s_i, \dot{s_i})$. We define

$$\begin{aligned} s_{i+1} &= \mathrm{Inf}\{s > s_i, \ddot{s_i} \notin [(1-2\mu)\beta(s, \Gamma_{(s_i,\dot{s_i},\ddot{s_i})}(s)) \\ &\qquad + 2\mu\alpha(s, \Gamma_{(s_i,\dot{s_i},\ddot{s_i})}(s)), \beta(s, \Gamma_{(s_i,\dot{s_i},\ddot{s_i})}(s))]\} \\ \dot{s}_{i+1} &= \Gamma_{(s_i,\dot{s_i},\ddot{s_i})}(s_{i+1}) \end{aligned}$$

The acceleration curve starting from $(s_0, \dot{s}_0)$ is then defined by $\dot{s} = \Gamma_{(s_i,\dot{s}_i,\ddot{s}_i)}(s)$ over each interval $[s_i, s_{i+1}]$ (Figure 41). Let notice that when μ tends to zero these curves tends to maximal velocity curves.

Deceleration curves are identically defined, replacing α by β.

Algorithm: Starting from $(s_{start}, 0)$, we build an acceleration curve until an out-point is reached. Then we build a deceleration curve backward from the next characteristic point. If the deceleration curve intersects the acceleration curve, we start again from the characteristic point. If the deceleration curve reaches the area above the last acceleration curve, it is stopped and the acceleration curve is extended. Finally the last deceleration is built backward from $(s_{end}, 0)$ until it intersects the already built curve. For more details we refer to [57].

The notion of in-points and out-points that we introduced, enables us to build only necessary curves and to stay as far as possible from the maximal velocity curve where the interval of admissible pseudo-accelerations is small. Slotine et al build curves from every characteristic point and take the lower envelop of all these curves.

Figure 42 shows an example of phase curve taking into account only the acceleration constraints.

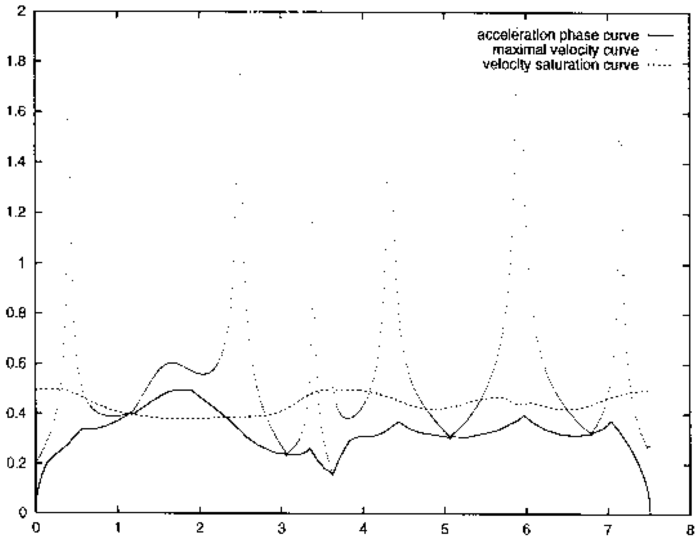


Figure 42: Phase curve taking into account the acceleration constraints

The velocity constraints: From now on, we call the formerly built phase plane curve the *acceleration phase curve*. From this curve we are going to build another one which takes into account the velocity constraints of the robot.

The method consists in following the acceleration phase curve until a velocity constraint is violated. Then the velocity saturation curve is followed as long as its slope corresponds to a suitable acceleration, and the acceleration phase curve remains above the velocity saturation curve. Three events can then occur:

1. The slope of the velocity saturation curve becomes too big: an acceleration curve is built until it reaches the velocity saturation again or until it reaches the acceleration phase curve.

2. The slope of the velocity saturation curve becomes too small: from the next point on this curve such that the slope is again suitable, a deceleration curve is built.

3. The acceleration phase curve intersects the velocity saturation curve: it is followed until it intersects again the velocity saturation curve.

Figure 43 shows the final phase curve taking into account all the constraints.

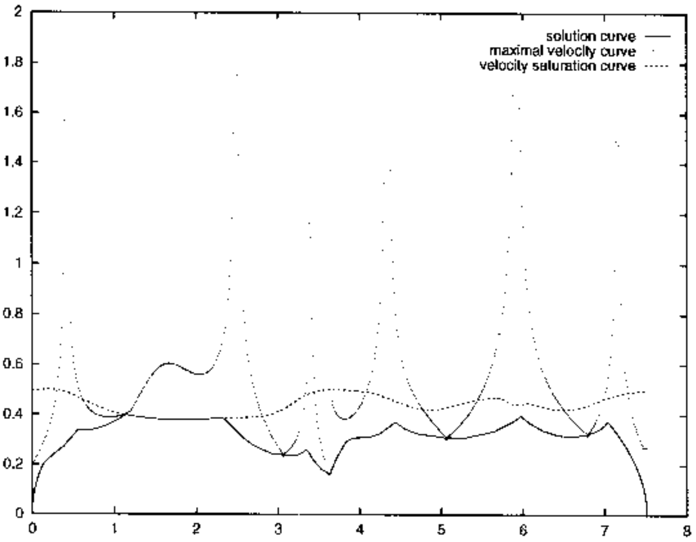


Figure 43: Final phase curve

E.3 Motion control

E.3.1 Motivation and related work

Motion control for nonholonomic systems have given rise to a lot of work. Brockett's condition [19] made stabilization about a given configuration a challenging task for such systems, proving that it could not be performed by a simple continuous state feedback. Alternative solutions as time-varying feedback [77, 28, 84, 93, 94, 95, 105] or discontinuous feedback [26] have been then proposed. On the other hand, tracking a trajectory for a nonholonomic system does not meet Brockett's condition and thus it is an easier task. A lot of work have also addressed this problem [31, 42, 52, 92, 96] for the particular case of mobile robots. See [29] for a recent survey in mobile robot motion control.

All these control laws work under the same assumption: the evolution of the system is exactly known and no perturbation makes the system deviate from its trajectory. Few papers dealing with mobile robots control take into account perturbations in the kinematics equations. [13] however proposed a method to stabilize a car about a configuration, robust to control vector fields perturbations, and based on iterative trajectory tracking.

The presence of obstacle makes the task of reaching a configuration even more difficult. The approach we have implemented combines iteratively open loop controls together with closed loop controls. Such a strategy is analyzed by assuming that the execution of a given trajectory is submitted to perturbations. The model we chose for these perturbations is simple and general. It presents some common points with [13].

E.3.2 Trajectory tracking

The low velocity (50 cm/s) of Hilare's motions, the good quality of its locomotion system and the good quality of the planned trajectories make the trajectory tracking task non critical. We devised a simple control law enabling us to reuse the controller of our robot without trailer. This controller was directly derived from [96]. Let (x, y, θ) be the coordinates of the reference robot in the frame of the real robot. Let (v_r^0, ω_r^0) be the inputs of the reference trajectory. The control law has the following expression:

$$\left\{\begin{array}{lcl} v_r & = & v_r^0 \cos\theta + k_1 x \\ \omega_r & = & \omega_r^0 + k_3\theta + k_2 \frac{\sin\theta}{\theta} y \end{array}\right. \tag{38}$$

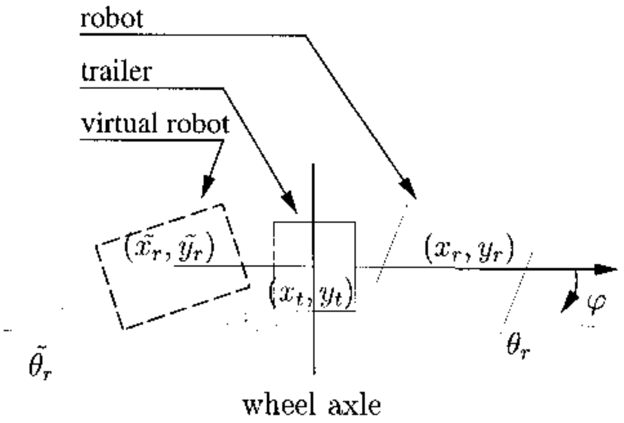


Figure 44: Virtual robot

System A: The idea of our controller is the following. When the robot goes forward, the trailer is not taken into account and we stabilize the robot according to the simple control law above. When the robot goes backward, we define a virtual robot which is symmetrical to the real robot with respect to the wheel axle of the trailer (Figure 44). The configuration of the virtual robot with respect to the real one is given by:

$$\begin{cases} \tilde{x_r} = x_r - 2\ell\cos\theta_t \\ \tilde{y_r} = y_r - 2\ell\sin\theta_t \\ \tilde{\theta}_r = \theta_l - \varphi + \pi \end{cases}$$

If $(\tilde{v}, \tilde{\omega})$ are the linear and angular velocities of the virtual robot, we get : $\tilde{v} = -v, \tilde{\omega} = \frac{2v}{\ell}\sin(\varphi) - \omega$. Thus the virtual robot goes forward and virtually pulls the trailer. We apply therefore the control law (38) to the virtual robot $(\tilde{x_r}, \tilde{y_r}, \tilde{\theta_r})$.

System B: When the trailer is hitched behind the robot, the former construction is even more simple: we can replace the virtual robot by the trailer. In this case indeed, the velocities of the robot (v_r, ω_r) and of the trailer (v_t, ω_t) are connected by a one-to-one mapping. The configuration of the virtual robot is then given by the following system:

$$\begin{cases} \tilde{x}_r &=& x_r - l_r\cos\theta_r - l_t\cos(\theta_r + \varphi) \\ \tilde{y}_r &=& y_r - l_r\sin\theta_r - l_t\sin(\theta_r + \varphi) \\ \tilde{\theta}_r &=& \theta_r + \varphi + \pi \end{cases}$$

Stability of the trailer: Do the previous approaches make the motion of the trailer truly stable ? To answer the question we consider here a forward motion for the system A. The analysis of backward motions is equivalent by considering the virtual robot transformation. Moreover the following analysis may be applied as well to system B by considering the motion of the hitching point.

Let us denote by $(x_r^0, y_r^0, \theta_r^0, \varphi^0, v_r^0, \omega_r^0)$ a reference trajectory and by $(x_r, y_r, \theta_r, \varphi, v_r, \omega_r)$ the real motion of the system. We assume that the robot follows exactly its reference trajectory: $(x_r, y_r, \theta_r, v_r, \omega_r) = (x_r^0, y_r^0, \theta_r^0, v_r^0, \omega_r^0)$ and we focus our attention on the trailer deviation $\hat{\varphi} = \varphi - \varphi^0$. The evolution of this deviation is easily deduced from System A ($l_r = 0$):

$$\begin{aligned} \dot{\hat{\varphi}} &= -\frac{v_r}{l_t}(\sin\varphi - \sin\varphi^0) \\ &= -\frac{2v_r}{\ell_t}\cos(\frac{\varphi + \varphi^0}{2})\sin(\frac{\hat{\varphi}}{2}) \end{aligned}$$

$|\hat{\varphi}|$ is thus decreasing iff

$$-\frac{\pi}{2} < \varphi^0 + \frac{\hat{\varphi}}{2} < \frac{\pi}{2} \qquad [2\pi] \tag{39}$$

Our system is moreover constrained by the inequalities

$$-\pi/2 < \varphi, \varphi^0 < \pi/2 \tag{40}$$

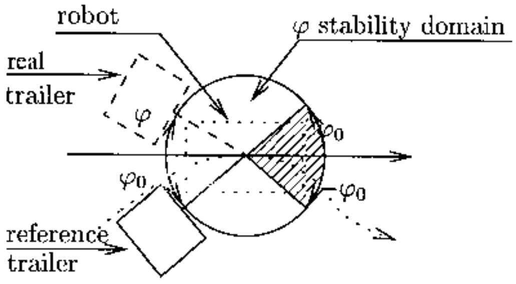


Figure 45: Stability domain for $\hat{\varphi}$

so that $-\pi < \hat{\varphi} < \pi$ and (39) is equivalent to

$$\left\{ \begin{array}{ccc} 0 < \varphi^0 < \frac{\pi}{2} & \text{and} & -\pi < \hat{\varphi} < \pi - 2\varphi^0 \\ & \text{or} & \\ -\frac{\pi}{2} < \varphi^0 < 0 & \text{and} & -\pi - 2\varphi^0 < \hat{\varphi} < \pi \end{array} \right. \tag{41}$$

Figure 45 shows the domain on which $|\hat{\varphi}|$ is decreasing for a given value of φ^0. We can see that this domain contains all positions of the trailer defined by the bounds (40). Moreover, the previous computations permit easily to show that 0 is an asymptotically stable value for the variable $\hat{\varphi}$.

Thus if the real or virtual robot follows its reference forward trajectory, the trailer is stable and will converge toward its own reference trajectory.

E.3.3 Iterative scheme and robustness

Once the robot stops after tracking a planned trajectory, the gap to the real goal is computed. If this gap is greater than some threshold, then a new trajectory is planned and tracked. As we will see below this simple iterative scheme gives good results. Usually, no more than one maneuver is needed to improve the final position of the system. Before presenting the experimental results, let us analyze the robustness of this control scheme from a formal point of view.

For this, we need to have a model of the perturbations arising when the robot moves. In our experiment we observed random perturbations due for instance to some play in the hitching system. These perturbations are very difficult to model. For this reason, we make only two simple hypotheses about them:

$$\begin{array}{rcl} \dot{d}_C(\dot{q}(s), \dot{q}^0(s)) & \leq & \delta s \\ d_C(q(s), q^0(s)) & \leq & \Delta \end{array}$$

where s is the curvilinear abscissa along the planned path, q and q^0 are respectively the real and reference configurations, d_C is a distance over the configuration space of the system and δ, Δ are positive constants. The first inequality means that the distance between the real and the reference configurations is proportional to the distance covered on the planned path. The second inequality is ensured by the trajectory tracking control law that prevents the system to go too far away from its reference trajectory. Let us point out that these hypotheses are very realistic and fit a lot of perturbation models.

We need now to know the length of the paths generated at each iteration. We have seen that the steering method we use to compute these paths is admissible. This means that if the goal is sufficiently close to the starting configuration, the computed trajectory remains in a neighborhood of the starting configuration. In [55] we compute an estimate in terms of distance: if q_1 and q_2 are two sufficiently close configurations, the length $\ell(q_1, q_2)$ of $\mathsf{Steer}_{flat}(q_1, q_2)$ verifies

$$\ell(q_1, q_2) < \eta d_C(q_1, q_2)^{\frac{1}{4}}$$

where η is a positive constant.

Thus, if $(q_i)_{i=1,2,\ldots}$ is the sequence of configurations reached after i motions, we have the following inequalities:

$$\begin{aligned} d_C(q_1, q_{goal}) &\leq \Delta \\ d_C(q_{i+1}, q_{goal}) &\leq \delta\ell(x_i, x_{goal}) \\ &\leq \delta\eta d_C(q_i, q_{goal})^{\frac{1}{4}} \end{aligned}$$

These inequalities ensure that $d_C(q_i, q_{goal})$ is upper bounded by a sequence $(d_i)_{i=1,2,\ldots}$ of positive numbers defined by

$$\begin{aligned} d_1 &= \Delta \\ d_{i+1} &= \delta\eta d_i^{\frac{1}{4}} \end{aligned}$$

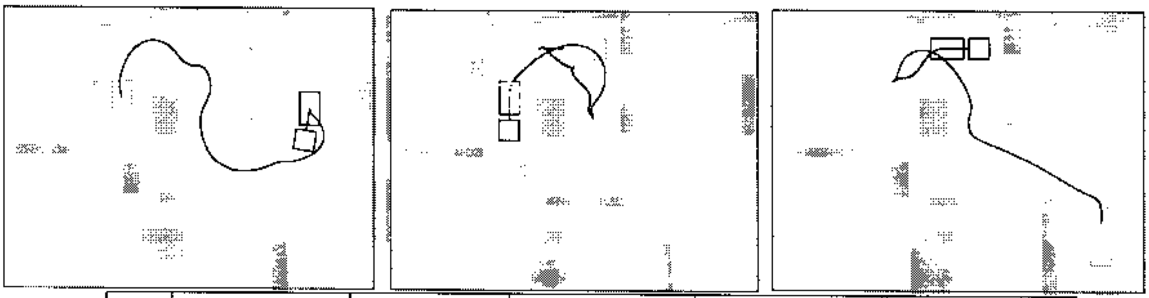
and converging toward $(\delta\eta)^{\frac{4}{3}}$.

Thus, we do not obtain asymptotical stability of the goal configuration, but this result ensures the existence of a stable domain around this configuration. This result essentially comes from the very general model of perturbations we have chosen. Let us repeat that including such a perturbation model in a time varying control law would undoubtedly make it lose its asymptotical stability.

The experimental results of the following section show however, that the converging domain of our control scheme is very small.

E.4 Experiments

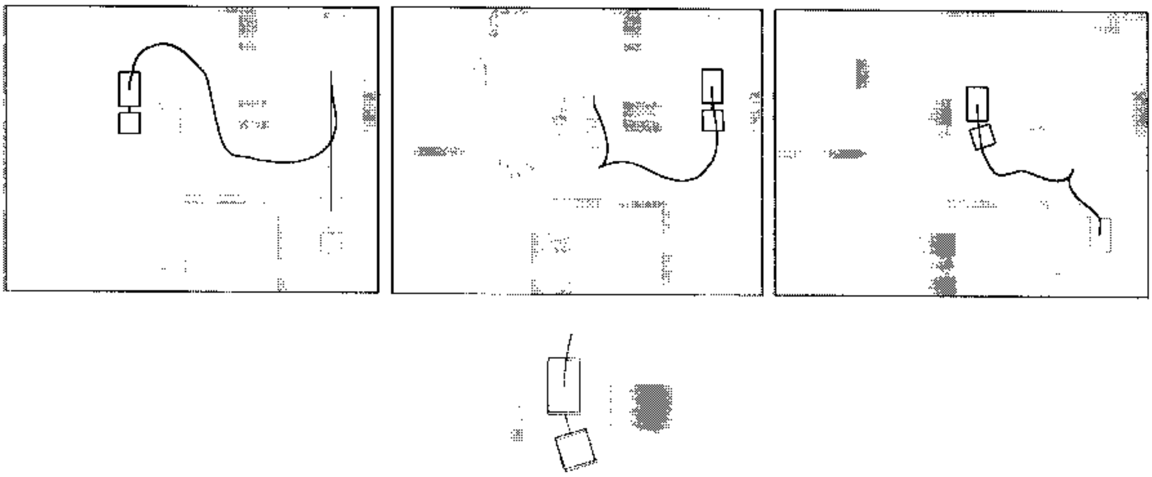
We present three experiments for each system. The geometric map of the environment covers 170 m^2. The bitmap representation of the environment is a grid of 150000 pixels. The geometric parameters of System A are $l_r = 0\ cm$ and $l_t = 120\ cm$. Those of System B are $l_r = 65\ cm$ and $l_t = 90\ cm$. For both systems, the bounds on the velocities and accelerations are $v_{max} = .5\ ms^{-1}$, $\omega_{max} = .5\ rads^{-1}$, $\dot{v}_{max} = .5\ ms^{-2}$ and $\dot{\omega}_{max} = 1.8\ rads^{-2}$.



	q_{init}	q_{but}	Δq_1	Δq_2	
I	$x = 9.42$ $y = -2.90$ $\theta = 91.39$ $\varphi = -10.19$	$x = 2.04$ $y = -2.5$ $\theta = 90$ $\varphi = 0$	$\Delta x = 0$ $\Delta y = -0.02$ $\Delta\theta = 0.18$ $\Delta\varphi = 0$		18 sec 0 cusp
II	$x = 2.10$ $y = -2.43$ $\theta = 89.7$ $\varphi = 0$	$x = 4.26$ $y = -0.50$ $\theta = 180$ $\varphi = 0$	$\Delta x = 0$ $\Delta y = 0.14$ $\Delta\theta = 4.13$ $\Delta\varphi = -4.93$	$\Delta x = 0.01$ $\Delta y = -0.02$ $\Delta\theta = 1.96$ $\Delta\varphi = -0.71$	1 min 53 sec 3 cusps
III	$x = 4.27$ $y = -0.51$ $\theta = 179.63$ $\varphi = 0.35$	$y = 10.30$ $y = -7.32$ $\theta = 90$ $\varphi = 0$	$\Delta x = 0.13$ $\Delta y = 0.01$ $\Delta\theta = 4.45$ $\Delta\varphi = -4.93$	$\Delta x = -0.02$ $\Delta y = -0.01$ $\Delta\theta = 0.96$ $\Delta\varphi = -0.35$	23 sec 1 cusp

Figure 46: Three experiments I, II and III from letf to right, for system A. The initial configuration is in black, the final configuration is in grey.

For each experiment, we proceed as follows. We localize the initial position of the robot using the cameras on the ceiling. Then we specify a goal configuration via the interface. After computations, the motion is executed. The position of the robot is updated by the dead-reckoning system combining the odometer of Hilare-2-bis and the angular encoder of the trailer. If the reached configuration is too far from the goal, we reexecute the same process. Figures 46 and 47 display the paths computed and give the precision reached after the first and second motions, which respect to the dead-reckoning localization. The times of computation correspond to the first path planning task on a Sun Sparc Ultra. The time parameterization is very fast (< 1 sec). Let us point out that the second planning task is almost instantaneous because both configurations are very close to one another and only one call to the local planner is generally enough. The exact position of the robot cannot be measured exactly after each motion because the robot is not always



	q_{init}	q_{but}	Δq_1	Δq_2	
I	$x = 2.20$ $y = -2.24$ $\theta = 90$ $\varphi = 0$	$x = 10.10$ $y = -6.90$ $\theta = 90$ $\varphi = 0$	$\Delta x = -0.05$ $\Delta y = 0.02$ $\Delta\theta = -1.10$ $\Delta\varphi = 1.41$	$\Delta x = 0$ $\Delta y = -0.01$ $\Delta\theta = -1.90$ $\Delta\varphi = 1.05$	1 min 10 sec 1 cusp
II	$x = 9.99$ $y = -2.00$ $\theta = 90.20$ $\varphi = 1.41$	$x = 5.34$ $y = -2.66$ $\theta = 90$ $\varphi = 18.43$	$\Delta x = 0$ $\Delta y = -0.06$ $\Delta\theta = 0.54$ $\Delta\varphi = 1.28$	$\Delta x = 0$ $\Delta y = -0.01$ $\Delta\theta = -0.75$ $\Delta\varphi = 0.58$	1 min 12 sec 1 cusp
III	$x = 5.31$ $y = -2.61$ $\theta = 89.44$ $\varphi = 19.01$	$x = 10.08$ $y = -7.70$ $\theta = -90$ $\varphi = 27.65$	$\Delta x = 0.02$ $\Delta y = 0.03$ $\Delta\theta = 0.81$ $\Delta\varphi = -0.19$		40 sec 1 cusp

Figure 47: Three experiments I, II and III from letf to right, for system B. The initial configuration is in black, the final configuration is in grey. The zoom shows final maneuver of the experiment II.

under one of the cameras. However, for paths such as those we executed in these experiments, the drift of the dead-reckoning system is less than 5 cm. We give the accuracy of the reached configuration only at the end of the motion because we experienced that the error during the motion increases at the beginning of the motion and then remains stable. Thus, values at the end are a good estimate of the precision during the whole motion.

Figure 46 and 47 gather the results for System A and System B respectively. Times correspond to the total time of the planning phase and the transformation of the paths into the trajectories to be executed.

References

[1] R. Alami, "Multi-robot cooperation based on a distributed and incremental plan merging paradigm," *Algorithms for Robotic Motion and Manipulation, WAFR'96*, J.P. Laumond and M. Overmars Eds, A.K. Peters, 1997.

[2] P.K. Agarwal, P. Raghavan and H. Tamaki, "Motion planning for a steering-constrained robot through moderate obstacles," *ACM Symp. on Computational Geometry*, 1995.

[3] J.M. Ahuactzin, " Le Fil d'Ariane: une méthode de planification générale. Application à la planification automatique de trajectoires," PhD Thesis, INP, Grenoble, 1994.

[4] F. Avnaim, J. Boissonnat and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles," *IEEE Int. Conf. on Robotics and Automation*, pp. 1656–1661, Philadelphia, 1988.

[5] J. Barraquand and J.C. Latombe, "Robot motion planning: a distributed representation approach," *Int. Journal of Robotics Research*, 1991.

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

[7] J. Barraquand and J.C. Latombe, "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles," *Algorithmica*, Springer Verlag, Vol 10, pp. 121–155, 1993.

[8] J. Barraquand, L. Kavraki, J.C. Latombe, T.Y. Li, R. Motvani and P. Raghavan, "A random sampling scheme for path planning," *Robotics Research, the Seventh Int. Symposium*, G. Giralt and G. Hirzinger Eds, Springer Verlag, 1996.

[9] A. Bellaïche, J.P. Laumond and P. Jacobs, "Controllability of car-like robots and complexity of the motion planning problem," *Int. Symposium on Intelligent Robotics*, pp. 322–337, Bangalore, 1991.

[10] A. Bellaïche, J.P. Laumond and J.J. Risler, "Nilpotent infinitesimal approximations to a control Lie algebra," *IFAC Nonlinear Control Systems Design Symposium*, pp. 174–181, Bordeaux, 1992.

[11] A. Bellaïche, J.P. Laumond and M. Chyba, "Canonical nilpotent approximation of control systems: application to nonholonomic motion

planning," *32nd IEEE Conf. on Decision and Control*, San Antonio, 1993.

[12] A. Bellaïche, F. Jean and J.J. Risler, "Geometry of nonholonomic systems," in *Robot Motion Planning and Control*, J.P. Laumond Ed., Lecture Notes in Control and Information Sciences, 229, Springer Verlag, 1998.

[13] M. K. Bennani and P. Rouchon, "Robust stabilization of flat and chained systems", *European Control Conf.*, 1995.

[14] P. Bessière, J.M. Ahuactzin, E. Talbi and E. Mazer, "The Ariadne's clew algorithm: global planning with local methods," *Algorithmic Foundations of Robotics*, K. Goldberg *et al* Eds, A.K. Peters, 1995.

[15] J.D. Boissonnat, A. Cerezo and J. Leblong, "Shortest paths of bounded curvature in the plane," *IEEE Conf. on Robotics and Automation*, pp. 2315–2320, Nice, France, 1992.

[16] J.D. Boissonnat and S. Lazard, "A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacle," *ACM Symp. on Computational Geometry*, 1996.

[17] N. Bourbaki, *Groupes et Algèbres de Lie : chapitres 2 et 3* CCLS Diffusion, Paris, 1972.

[18] R.W. Brockett, "Control theory and singular Riemannian geometry," *New Directions in Applied Mathematics*, Springer-Verlag, 1981.

[19] R.W. Brockett, "Asymptotic stability and feedback stabilization", in *Differential Geometric Control Theory*, R.W. Brockett, R.S. Millman and H.H. Sussmann Eds, 1983.

[20] X.-N. Bui, P. Souères, J.-D. Boissonnat and J.-P. Laumond, "Shortest path synthesis for Dubins non-holonomic robots," *IEEE Int. Conf. on Robotics and Automation*, pp. 2–7, San Diego, CA, May 1994.

[21] X.-N. Bui, "Planification de trajectoire pour un robot polygonal nonholonome dans un environnement polygonal," Ph.D. Thesis, Ecole Nationale Supérieure des Mines de Paris, 1994.

[22] L. Bushnell, D. Tilbury and S. Sastry, "Steering three-input nonholonomic systems: the fire-truck example," *Int. Journal of Robotics Research*, Vol 14 (4), pp. 366–381, 1995.

[23] G. Campion, G. Bastin and B. d'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. on Robotics and Automation*, Vol 12 (1), 1996.

[24] J. Canny, *The Complexity of Robot Motion Planning*, MIT Press, 1988.

[25] J. Canny, A. Rege and J. Reif, "An exact algorithm for kinodynamic planning in the plane," *Discrete and Computational Geometry*, Vol 6, pp. 461–484, 1991.

[26] C. Canudas de Wit, O.J. Sørdalen, "Exponential stabilization of mobile robots with non holonomic constraints," *IEEE Transactions on Automatic Control*, Vol 37, No. 11, 1992.

[27] E. J. Cockayne and G. W. C. Hall, "Plane motion of a particle subject to curvature constraints," *SIAM J. on Control*, **13** (1), 1975.

[28] J. M. Coron, "Global asymptotic stabilization for controllable systems without drift," *Mathematics of Control, Signals and Systems*, Vol 5, 1992.

[29] A. De Luca, G. Oriolo and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*, J.P. Laumond Ed., Lecture Notes in Control and Information Sciences, Springer Verlag, 229, 1998.

[30] B. Donald, P. Xavier, J. Canny and J. Reif, "Kinodynamic motion planning," *J. of the ACM*, Vol 40, pp. 1048–1066, 1993.

[31] R. M. DeSantis, "Path-tracking for a tractor-trailer-like robot," *Int. Journal of Robotics Research*, Vol 13, No 6, 1994.

[32] B. Donald and P. Xavier, "Provably good approximation algorithms for optimal kinodynamic planning: robots with decoupled dynamic bounds," *Algorithmica*, Vol 14, pp. 443–479, 1995.

[33] 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*, Vol 79, pp. 497–516, 1957.

[34] P. Ferbach, "A method of progressive constraints for nonholonomic motion planning," *IEEE Int. Conf. on Robotics and Automation*, pp. 2929–2955, Minneapolis, 1996.

[35] C. Fernandes, L. Gurvits and Z.X. Li, "A variational approach to optimal nonholonomic motion planning," *IEEE Int. Conf. on Robotics and Automation*, pp. 680–685, Sacramento, 1991.

[36] S. Fleury, P. Souères, J.P. Laumond and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Transactions on Robotics and Automation*, Vol 11 (3), pp. 441–448, 1995.

[37] M. Fliess, J. Lévine, P. Martin and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *Int. Journal of Control*, Vol 61 (6), pp. 1327–1361, 1995.

[38] S.J. Fortune and G.T. Wilfong, "Planning constrained motions," *ACM STOCS*, pp. 445–459, Chicago, 1988.

[39] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1393–1400, Yokohama, 1993.

[40] V. Y. Gershkovich, "Two-sided estimates of metrics generated by absolutely nonholonomic distributions on Riemannian manifolds," *Soviet Math. Dokl.*, Vol 30 (2), 1984.

[41] G. Giralt, R. Sobek and R. Chatila, "A multi-level planning and navigation system for a mobile robot: a first approach to Hilare," *6th Int. Joint Conf. on Artificial Intelligence*, pp. 335–337, Tokyo, 1979.

[42] A. Hemami, M. G. Mehrabi and R. M. H. Cheng, "Synthesis of an optimal control law path tracking in mobile robots," *Automatica*, Vol 28, No 2, pp 383–387, 1992.

[43] H. Hermes, A. Lundell and D. Sullivan, "Nilpotent bases for distributions and control systems," *J. of Differential Equations*, Vol 55, pp. 385–400, 1984.

[44] J. Hopcroft, J.T. Schwartz and M. Sharir, "On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the warehouseman's problem," *Int. Journal of Robotics Research*, Vol 3, pp. 76–88, 1984.

[45] G. Jacob, "Lyndon discretization and exact motion planning," *European Control Conf.*, pp. 1507–1512, Grenoble, 1991.

[46] P. Jacobs and J. Canny, " Planning smooth paths for mobile robots," *IEEE Int. Conf. on Robotics and Automation*, Scottsdale, 1989.

[47] P. Jacobs, A. Rege and J.P. Laumond, "Non-holonomic motion planning for Hilare-like robots," *Int. Symposium on Intelligent Robotics*, pp. 338–347, Bangalore, 1991.

[48] James and James, *Mathematics Dictionary*, Princeton, NJ: Van Nostrand, 1968.

[49] H. Jaouni, M. Khatib and J.P. Laumond, "Elastic band for nonholonomic car-like robots: algorithms and combinatorial issues," *Workshop on Algorithmic Foundations of Robotics*, Houston, 1998.

[50] F. Jean, "The car with N trailers: characterization of the singular configurations," *ESAIM: COCV*, http://www.emath.fr/cocv/, Vol 1, pp. 241–266, 1996.

[51] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," *Robotics Research*, Vol 3, MIT Press, pp. 333–340, 1986.

[52] Y. Kanayama, Y. Kimura, F. Miyazaki and T.Nogushi, "A stable tracking control method for an autonomous mobile robot," *IEEE Int. Conf. on Robotics and Automation*, Cincinnati, Ohio, 1990.

[53] M. Khatib, H. Jaouni, R. Chatila and J.P. Laumond, " Dynamic path modification for car-like nonholonomic mobile robots," *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, 1997.

[54] G. Lafferriere and H.J. Sussmann, "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning*, Zexiang Li and J.F. Canny Eds, The Kluwer International Series in Engineering and Computer Science 192, 1992.

[55] F. Lamiraux and J.P. Laumond, "From paths to trajectories for multibody mobile robots," *Int. Symposium on Experimental Robotics*, Lecture Notes on Control and Information Science, Springer-Verlag, (to appear) 1997.

[56] F. Lamiraux and J.P. Laumond, "Flatness and small-time controllability of multi-body mobile robots: applications to motion planning," *European Conf. on Control*, Brussels, 1997.

[57] F. Lamiraux and J.P. Laumond, "From paths to trajectories for multi-body mobile robots", *Int. Symposium on Experimental Robotics*, Barcelona, 1997.

[58] F. Lamiraux, "Robots mobiles à remorques : de la planification de chemins à l'exécution de mouvements," PhD Thesis 1329, INPT, LAAS-CNRS, Toulouse, 1997.

[59] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[60] J.C. Latombe, "A Fast Path Planner for a Car-Like Indoor Mobile Robot," *Ninth National Conf. on Artificial Intelligence, AAAI*, pp. 659–665, Anaheim, 1991.

[61] J.P. Laumond, "Feasible trajectories for mobile robots with kinematic and environment constraints," *Intelligent Autonomous Systems*, L.O. Hertzberger, F.C.A. Groen Eds, North-Holland, pp. 346–354, 1987.

[62] J.P. Laumond, "Finding collision-free smooth trajectories for a nonholonomic mobile robot," *10th International Joint Conference on Artificial Intelligence*, pp. 1120–1123, Milano, 1987.

[63] J.P. Laumond, T. Siméon, R. Chatila and G. Giralt, "Trajectory planning and motion control for mobile robots," *Geometry and Robotics, J.D. Boissonnat and J.P. Laumond (Eds)*, pp. 133–149, Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.

[64] J.P. Laumond and T. Siméon, "Motion planning for a two degree of freedom mobile robot with towing," LAAS/CNRS Report 89148, 1989.

[65] J.P. Laumond, " Singularities and topological aspects in nonholonomic motion planning," in *Nonholonomic Motion Planning*, Zexiang Li and J.F. Canny Eds, The Kluwer International Series in Engineering and Computer Science 192, 1992.

[66] J.P. Laumond, "Controllability of a Multibody Mobile Robot," *IEEE Transactions Robotics and Automation*, pp. 755–763, Vol 9 (6), 1993.

[67] J.P. Laumond, P. Jacobs, M. Taïx and R. Murray, "A motion planner for nonholonomic mobile robot," *IEEE Trans. on Robotics and Automation*, Vol 10, 1994.

[68] J.P. Laumond, S. Sekhavat and M. Vaisset, "Collision-free motion planning for a nonholonomic mobile robot with trailers," *4th IFAC Symp. on Robot Control*, pp. 171–177, Capri, 1994.

[69] J.P. Laumond and J.J. Risler, "Nonholonomic systems: controllability and complexity," *Theoretical Computer Science*, Vol 157, pp. 101–114, 1996.

[70] J.P. Laumond and P. Souères, "Metric induced by the shortest paths for a car-like robot," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Yokoama, 1993.

[71] Z. Li and J.F. Canny Eds, *Nonholonomic Motion Planning*, Kluwer Academic Publishers, 1992.

[72] T. Lozano-Pérez, "Spatial planning: a configuration space approach," *IEEE Trans. Computers*, Vol 32 (2), 1983.

[73] F. Luca and J.J. Risler, "The maximum of the degree of nonholonomy for the car with n trailers," *IFAC Symp. on Robot Control*, pp. 165–170, Capri, 1994.

[74] B. Mirtich and J. Canny, "Using skeletons for nonholonomic path planning among obstacles," *IEEE Int. Conf. on Robotics and Automation*, Nice, 1992.

[75] J. Mitchell, "On Carnot-Carathéodory metrics," *J. Differential Geometry*, Vol 21, pp. 35-45, 1985.

[76] S. Monaco and D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," *IEEE Int. Conf. on Decision and Control*, pp. 1780–1785, Tucson, 1992.

[77] P. Morin and C. Samson, *Application of backstepping techniques to the time-varying exponential stabilisation of chained form systems*, European Journal of Control, Vol 3, No 1, 1997.

[78] P. Moutarlier, B. Mirtich and J. Canny, Shortest paths for a car-like robot to manifolds in configuration space, *Int. Journal of Robotics Research*, **15** (1), 1996.

[79] R.M. Murray and S. Sastry, "Steering nonholonomic systems using sinusoids," *IEEE Int. Conf. on Decision and Control*, pp. 2097–2101, 1990.

[80] R.M. Murray, "Robotic Control and Nonholonomic Motion Planning," PhD Thesis, Memorandum No. UCB/ERL M90/117, University of California, Berkeley, 1990.

[81] R.M. Murray, "Nilpotent bases for a class on nonintegrable distributions with applications to trajectory generation for nonholonomic systems," *Math. Control Signal Syst.*, Vol 7, pp. 58–75, 1994.

[82] N.J. Nilsson, "A mobile automaton: an application of artificial intelligence techniques," *1st Int. Joint Conf. on Artificial Intelligence*, pp. 509–520, Washington, 1969.

[83] C. O'Dunlaing, "Motion planning with inertial constraints," *Algorithmica*, Vol 2 (4), 1987.

[84] J. B. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Systems and Control Letters*, North Holland, Vol 18, pp 147–158, 1992.

[85] P. Rouchon, M. Fliess, J. Lévine and P. Martin, "Flatness and motion planning: the car with n trailers," *European Control Conf.*. pp. 1518–1522, 1993.

[86] P. Rouchon, "Necessary condition and genericity of dynamic feedback linearization," in *J. Math. Systems Estimation Control*, Vol 4 (2), 1994.

[87] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific Journal of Mathematics*, 145 (2), pp. 367–393, 1990.

[88] J. Reif and H. Wang, "Non-uniform discretization approximations for kinodynamic motion planning and its applications," *Algorithms for Robotic Motion and Manipulation, WAFR'96*, J.P. Laumond and M. Overmars Eds, A.K. Peters, 1997.

[89] J. Reif and H. Wang, "The complexity of the two dimensional curvature-constrained shortest-path problem," *Workshop on Algorithmic Foundations of Robotics*, Houston, 1998.

[90] M. Renaud and J.Y. Fourquet, "Time-optimal motions of robot manipulators including dynamics," *The Robotics Review 2*, O. Khatib, J.J. Craig and T. Lozano-Pérez Eds, MIT Press, 1992.

[91] J.J. Risler, "A bound for the degree of nonholonomy in the plane," *Theoretical Computer Science*, Vol 157, pp. 129–136, 1996.

[92] M. Sampei, T. Tamura, T Itoh and M. Nakamichi, "Path tracking control of trailer-like mobile robot," *IEEE Int. Workshop on Intelligent Robots and Systems*, Osaka, Japan, pp 193–198, 1991.

[93] C. Samson, "Velocity and torque feedback control of a nonholonomic cart," *Int. Workshop in Adaptative and Nonlinear Control: Issues in Robotics*, Grenoble, France, 1990.

[94] C. Samson, "Time-varying feedback stabilisation of car-like wheeled mobile robots," *Int. Journal of Robotics Research*, 12(1), 1993.

[95] C. Samson, "Control of chained systems. Application to path following and time-varying point-stabilization," *IEEE Transactions on Automatic Control*, Vol 40, No 1, 1995.

[96] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," *IEEE Int. Conf. on Robotics and Automation*, Sacramento, California, pp 1136–1141, 1991.

[97] J.T. Schwartz and M. Sharir, "On the 'Piano Movers' problem II: general techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, 4, pp. 298–351, 1983.

[98] S. Sekhavat, P. Švestka, J.P. Laumond and M. H. Overmars, "Multilevel path planning for nonholonomic robots using semi-holonomic subsystems," *Algorithms for Robotic Motion and Manipulation, WAFR'96*, J.P. Laumond and M. Overmars Eds, A.K. Peters, 1997.

[99] S. Sekhavat, "Planification de mouvements sans collisions pour systèmes non holonomes," PhD Thesis 1240, INPT, LAAS-CNRS, Toulouse, 1996.

[100] S. Sekhavat and J-P. Laumond, "Topological Property of Trajectories Computed from Sinusoidal Inputs for Chained Form Systems," *IEEE Int. Conf. on Robotics and Automation*, Mineapollis, 1996.

[101] S. Sekhavat and J-P. Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE Transaction on Robotics and Automation* (to appear).

[102] S. Sekhavat, F. Lamiraux, J-P. Laumond, G. Bauzil and A. Ferrand, "Motion planning and control for Hilare pulling a trailer: experimental issues," *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, 1997.

[103] Z. Shiller and Y. R. Gwo, "Dynamic motion planning of autonomous vehicles", *IEEE Transactions on Robotics and Automation*, 7(2), pp 241–249, 1991.

[104] J.J.E. Slotine and H.S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, 5 (1), pp.118-124, 1989.

[105] O.J. Sørdalen and O. Egeland, "Exponential stabilisation of nonholonomic chained systems," *IEEE Transactions on Automatic Control*, Vol 40, No 1, 1995.

[106] P. Souères and J.P. Laumond, "Shortest path synthesis for a car-like robot," *IEEE Trans. on Automatic Control*, Vol 41 (5), pp. 672–688, 1996.

[107] P. Souères, J.-Y. Fourquet and J.-P. Laumond, "Set of reachable positions for a car," *IEEE Trans. on Automatic Control*, 1994.

[108] E.D. Sontag, "Controllability is harder to decide than accessibility," *SIAM J. Control and Optimization*, Vol 26 (5), pp. 1106–1118, 1988.

[109] O.J. Sordalen, "Conversion of a car with n trailers into a chained form," *IEEE Int. Conf. on Robotics and Automation*, pp. 382–387, Atlanta, 1993.

[110] S. Sternberg, *Lectures on Differential Geometry*, Chelsea Pub., 1983.

[111] R. S. Strichartz, "Sub-Riemannian geometry," *Journal of Differential Geometry*, Vol 24, pp. 221–263, 1986.

[112] R. S. Strichartz, "The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations," *Journal of Functional Analysis*, Vol 72, pp. 320–345, 1987.

[113] H.J. Sussmann and V. Jurdjevic, " Controllability of nonlinear systems," *J. of Differential Equations*, 12, pp. 95–116, 1972.

[114] H. Sussmann, "Lie brackets, real analyticity and geometric control," *Differential Geometric Control Theory* (R. Brockett, R. Millman and H. Sussmann, eds.), Vol 27 of *Progress in Mathematics*, pp. 1–116, Michigan Technological University, Birkhauser, 1982.

[115] H. J. Sussmann and W. Liu, "Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories," Tech. Rep. SYSCON-91-02, Rutgers Center for Systems and Control, 1991.

[116] H.J. Sussmann and W. Tang, "Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control," Report SYCON-91-10, Rutgers University, 1991.

[117] P. Svestka and M. Overmars, "Coordinated motion planning for multiple car-like robots using probabilistic roadmaps," *IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, 1995.

[118] P. Svestka and M. Overmars, "Probabilistic path planning," in *Robot motion planning and control*, J.P. Laumond Ed., Lecture Notes in Control and Information Sciences, Springer Verlag, 229, 1998.

[119] D.Tilbury, R. Murray and S. Sastry, "Trajectory generation for the n-trailer problem using Goursat normal form," *IEEE Trans. on Automatic Control*, Vol 40 (5), pp. 802–819, 1995.

[120] D. Tilbury, J.P. Laumond, R. Murray, S. Sastry and G. Walsh, "Steering car-like systems with trailers using sinusoids," in *IEEE Conf. on Robotics and Automation*, pp. 1993-1998, Nice, 1992.

[121] A. Thompson, "The navigation system of the JPL robot," *5th Int. Joint Conf. on Artificial Intelligence*, pp. 749–757, Cambridge, 1977.

[122] P. Tournassoud, " Motion planning for a mobile robot with a kinematic constraint," *Geometry and Robotics*, J.D. Boissonnat and J.P. Laumond Eds, pp. 150–171, Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.

[123] V.S. Varadarajan, *Lie Groups, Lie Algebra and their Representations*, Springer-Verlag, 1984.

[124] M. Vendittelli and J.P. Laumond, "Visible positions for a car-like robot amidst obstacles," *Algorithms for Robotic Motion and Manipulation*, J.P. Laumond and M. Overmars Eds, A.K. Peters, 1997.

[125] M. Vendittelli, J.P. Laumond and G. Oriolo, "Nilpotent approximation of nonholonomic systems with singularities: a case of study," *IFAC Nonlinear Control Systems Design Symposium*, Enschede, the Netherlands, 1998.

[126] A.M. Vershik and V.Ya. Gershkovich, "Nonholonomic problems and the theory of distributions," *Acta Applicandae Mathematicae*, Vol 12, pp. 181–209, 1988.

[127] X. Viennot, "Algèbres de Lie libres et monoïdes libres," *Lecture Notes in Mathematics*, 691, Springer Verlag, 1978.

[128] G.T. Wilfong, " Motion planning for an autonomous vehicle," *IEEE Int. Conf. on Robotics and Automation*, pp. 529–533, 1988.