# Guidelines in Nonholonomic Motion Planning for Mobile Robots

J.P. Laumond S. Sekhavat F. Lamiraux

This is the first chapter of the book:

**Robot Motion Planning and Control**
Jean-Paul Laumond (Editor)

Laboratoire d'Analye et d'Architecture des Systèmes
Centre National de la Recherche Scientifique
LAAS report 97438

Previoulsy published as:
Lectures Notes in Control and Information Sciences 229.
Springer, ISBN 3-540-76219-1, 1998, 343p.

# Guidelines in Nonholonomic Motion Planning for Mobile Robots

J.P. Laumond, S. Sekhavat and F. Lamiraux

LAAS-CNRS, Toulouse

## 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 [61]; in 1977 Jason used a visibility graph built from the corners of the obstacles [88]; in 1979 Hilare decomposed its environment into collision-free convex cells [30].

At the end of the 70's the studies of robot manipulators popularized the notion of configuration space of a mechanical system [53]; 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 [42]).

One more decade, and the notion of nonholonomy (also borrowed from Mechanics) appears in the literature [44] 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 [52].

This chapter gives 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.

The chapter may be considered as self-contained; nevertheless, the basic necessary concepts in differential geometric control theory are more developed in Bellaïche–Jean–Risler's chapter.

Finally, notice that Nonholonomic Motion Planning may be consider as the problem of planning *open loop* controls; the problem of the feedback control is the purpose of DeLuca–Oriolo–Samson's chapter.

## 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 $\Sigma$ 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 [83].

**Definition 1.** *$\Sigma$ 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 [82]) 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 2.1.** *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. An algorithm appears in [46,50].

## 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 converge to the same modeling (e.g., [47]) but the later enlightens on practical issues more clearly than the former.

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

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

Let us consider two systems: a two-driving wheel mobile robot and a car (in [17] other mechanical structure of mobile robots are considered).

**Two-driving wheel mobile robots** A classical locomotion system for mobile robot is constituted by two parallel driving wheels, the acceleration of each being controlled by an independent motor. 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 $\theta$ of the driving wheels. With $\ell$ designating the distance between the driving wheels the dynamic model is:

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

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:

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

The bounds on $v_1$ and $v_2$ induce bounds $v_{max}$ and $\omega_{max}$ on the new controls $v$ and $\omega$. This system is symmetric without drift; applying the LARC condition shows that it is small-time controllable from everywhere. Notice that $v$ and $\omega$ should be $C^1$.

**Car-like robots** 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. 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 $\zeta$ as the angle between the front wheels and the main direction $\theta$ of the car[2]. Moreover a mechanical constraint imposes $|\zeta| \leq \zeta_{max}$ and consequently a

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

minimum turning radius. Simple computation shows that the dynamic model of the car is:

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

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.

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 $\omega$ 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 $\omega$ 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 [65] and [22] on the shortest paths in the plane with bounded curvature such systems will be called Reeds&Shepp's car and Dubins' car respectively (see Souères–Boissonnat's chapter 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[3]. 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 the car of Reeds & Shepp and the local controllability of Dubins' car may be illustrated geometrically. Figure 1 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 [76,51,15] and Souères–Boissonnat's chapter). 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.

### 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 2 (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 [47]) :

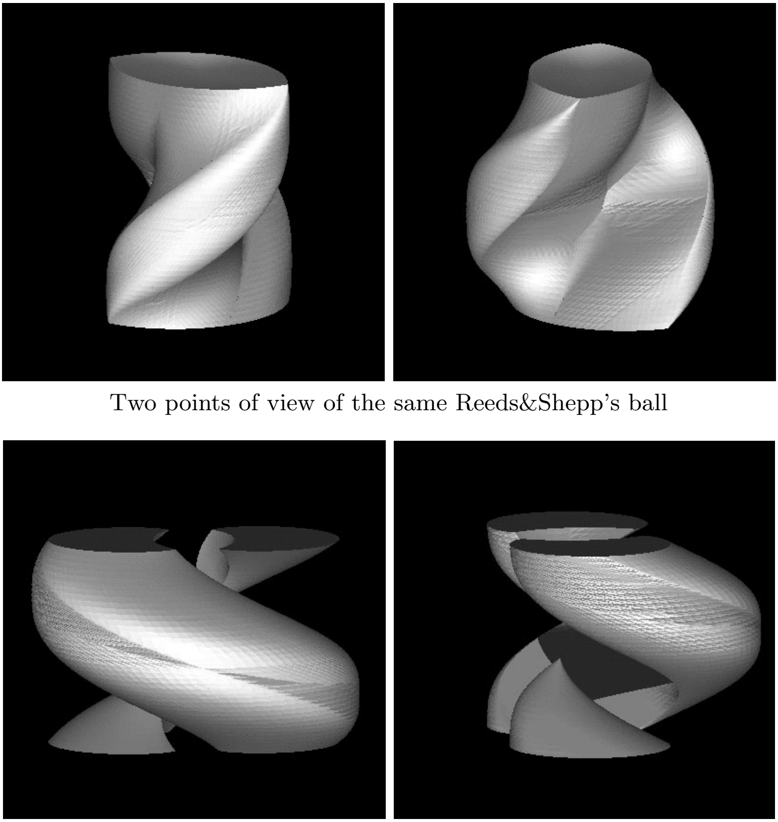
$$\dot{X} \;=\; f_1(X)v \;+ f_2(X)\omega \qquad (6)$$

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.

After noticing that $[f_2, [f_1, f_2]] = f_1$, one may check that the family composed of $\{f_1, f_2, [f_1, f_2], [f_1, [f_1, f_2]], [f_1, [f_1, [f_1, f_2]]]\}$ spans the tangent space at every point in $\mathbf{R}^2 \times (S^1)^3$ verifying $\varphi_1 \neq \frac{\pi}{2}$ (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

[3] A geometric proof appears in [44].



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

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

**Fig. 1.** Accessibility domains by shortest paths of fixed length

system is small-time controllable at any point. Its degree of nonholonomy[4] 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 [47].

Another hooking system is illustrated in Figure 2 (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

[4] 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 Bellaïche–Jean–Risler's Chapter for details).

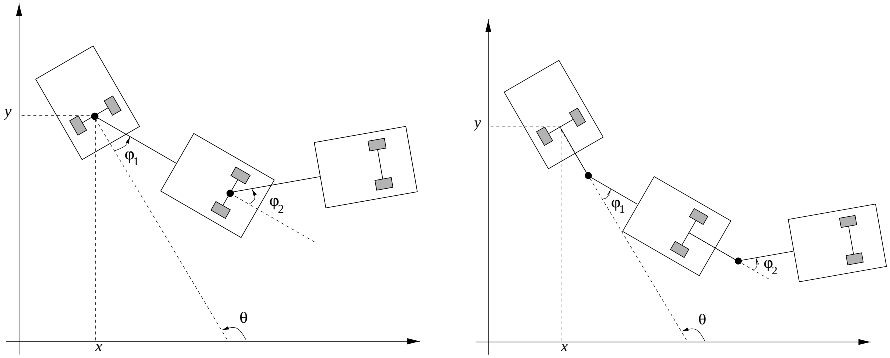


**Fig. 2.** Two types of mobile robots with trailers.

$f_1(X) = (\cos\theta,\ \sin\theta,\ 0,\ -\sin\varphi_1,\ -\sin\varphi_2\cos\varphi_1 + \cos\varphi_2\sin\varphi_1 + \sin\varphi_1)^T$ 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.

### 2.4 Admissible paths and trajectories

**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 3). 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).

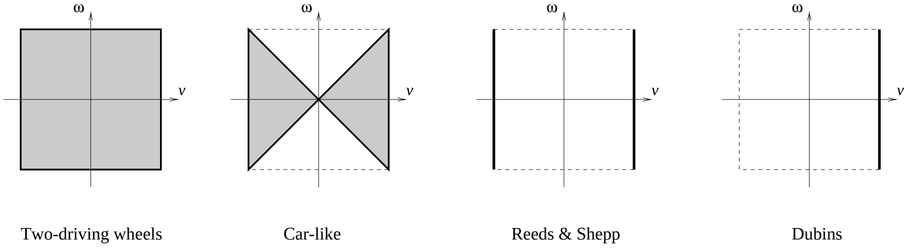


**Fig. 3.** Kinematic mobile robot models: four types of control domains.

*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 $\zeta$. Therefore the original control constraint $|\omega| < |v|$ arising in system (5) can be addressed by applying "obstacle" avoidance techniques to the system (4).

**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) [35]; they may be used to smooth elementary paths [25].

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., [67] for a survey of various approaches). Formal solutions exist (e.g., [75] for an approach using optimal control); they apply to our problem. Nevertheless, their practical programming tread on delicate numerical computations [40].

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., [20,21,66]). 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 [62,19]) and are time-consuming. Only few of them report results dealing with obstacle avoidance for nonholonomic mobile robots (e.g., [28]).

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 [27]. An approximated solution of the complete problem exists [34]; 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) [2,13]. 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 2 (left) we will see that the complete problem is solved; it remains open for the system in Figure 2 (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[5]. 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 positive real number $\eta$ such that the ball $B(X, \eta)$ centered at $X$ with radius $\eta$ is included in $\mathcal{R}each_X(T)$.

Now, let us consider a (not necessarily admissible) collision-free path $\gamma$ with finite length linking two configurations $X_{start}$ and $X_{goal}$. $\gamma$ being compact, it is possible to define the clearance $\epsilon$ of the path as the minimum distance of $\gamma$ to the obstacles[6]. $\epsilon$ is strictly positive. Then for any $X$ on $\gamma$, there exists $T_X > 0$ such that $\mathcal{R}each_X(T_X)$ does not intersect any obstacle. Let $\eta_{T_X}$ 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 $\gamma$. $\gamma$ being compact, it is possible to get a finite sequence of configurations $(X_i)_{1 \le i \le k}$ (with $X_1 = X_{start},\ \ X_k = X_{goal}$), such that the balls $B(X_i, \eta_{T_{X_i}})$ cover $\gamma$.

Consider a point $Y_{i,i+1}$ lying on $\gamma$ 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

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

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

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 3.1.** *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 3.1 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 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 [50]. The combinatorial complexity of the problem is addressed in [77]. Explicit bounds of complexity have been recently provided for polynomial systems in the plane (see [68] and references therein).

*Consequence 2:* Theorem 3.1 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 3.1 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 [79,90,80,81] and in Bellaïche–Jean–Risler's Chapter. 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 $\alpha$ being a real number, applying $e^{\alpha f}$ amounts to follow $f$ for a time $\alpha$. 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 $\alpha f$, then $\beta g$, 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 $\alpha f$ first (indeed $g$ is not evaluated at the same points in both cases).

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

The Campbell-Baker-Hausdorff-Dynkin formula states precisely the difference between the two systems: for a sufficiently small $\tau$, 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 $\epsilon(\tau) \rightarrow 0$ when $\tau \rightarrow 0$.

Actually, the whole formula as proved in [90] gives an explicit form for the $\epsilon$ function. More precisely, $\epsilon$ yields a formal series whose coefficients $c_k$ of $\tau^k$ are combinations of brackets of degree $k$,[7] 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. It 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 [39] 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.

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 for instance [46]). The method assumes that a holonomic path $\gamma$ 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 $\gamma$. 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. An introduction to this machinery through the example of a car-like robot appears in [48]. 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

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

in an iterated algorithm that produces a path ending as close to the goal as wanted.

In [33], 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 [93] 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 [31]). 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 [92]) 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[8]. A nilpotentization of this system appear in [39]. We conclude this section by the nilpotentization of a mobile robot pulling a trailer [11].

*Example:* Let us consider the control system 6:

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

where $(x, y)$ defines the position of the mobile robot, $\theta$ its direction and $\varphi$ 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

[8] 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 $ad_{f_2}^{2m}(f_1) = (-1)^m f_1 \neq 0$.

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

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

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

$$\widehat{f_1} = \begin{pmatrix} 1 \\ 0 \\ -z_2 \\ \widehat{F}(z_1, z_2, z_3) \end{pmatrix} \quad \widehat{f_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

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 [58,59] 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:

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

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 $i^{\text{th}}$ 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 [86] 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 [57].

Even if a system is not triangular, it may be possible to transform it into a triangular one by feedback transformations (see [59,60]). 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 2, 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 [86]. Denoting by $\theta_1$ and $\theta_2$ 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:

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

Let us consider the following change of coordinates:

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

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 [78].

*Sinusoidal inputs:* Let us consider the following inputs [86]:

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

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 4 from [71] illustrates this dependence. Moreover the shape of the paths is not invariant by rotation (i.e., it depends on the variables $\theta^{start}$ and $\theta^{goal}$ and not only on the difference $(\theta^{start} - \theta^{goal})$).

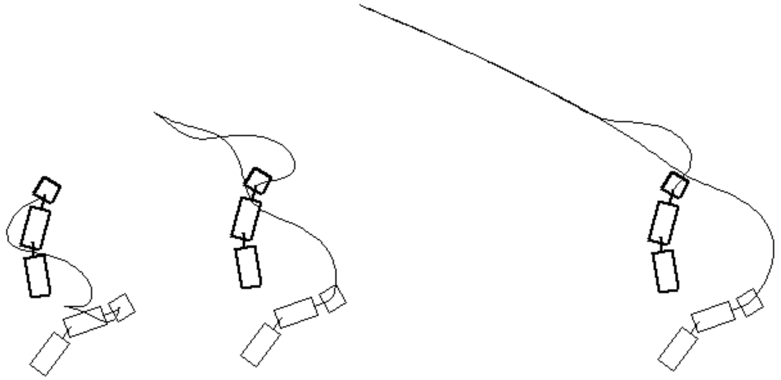


**Fig. 4.** Three paths solving the same problem with three values of $a_1$: -30, 70, 110

*Polynomial inputs:* Another steering method is also proposed in [86]. 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.

*Extensions:* The previous steering methods deal with two-input chained form systems. In [16] Bushnell, Tilbury and Sastry extend these results to three-input nonholonomic systems with the fire-truck system as a canonical example[9]. 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 [26,63].

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

Let us consider the system (6) (Figure 2, left). Let us denote the coordinates of the robot and the two trailers by $(x, y, \theta)$, $(x_1, y_1, \theta_1)$ and $(x_2, y_2, \theta_2)$ 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 $x_2$, $y_2$, $\theta_1$ and $\theta_2$:

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

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 $\theta_2$ (resp. $\theta_1$ and $\theta$) 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 $\omega$ 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.

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

*A steering method:* Let us consider a path $\gamma_2$ followed by the reference point $P_2$ of the second trailer (Figure 5). $\gamma_2$ is parametrized by arc length $s_2$. Let us assume that $\gamma_2$ is sufficiently smooth, i.e., $\frac{d}{ds_2}P_2$ is defined everywhere and the curvature $\kappa_2$ can be differentiated at least once. The point $P_1$ belongs to the tangent to $\gamma_2$ at $P_2$ and $P_1 = P_2 + \boldsymbol{\tau_2}$, with $\boldsymbol{\tau_2}$ the unitary tangent vector to $\gamma_2$. Differentiating this relation w.r.t. $s_2$ leads to $\frac{d}{ds_2}P_1 = \boldsymbol{\tau_2} + \kappa_2\boldsymbol{\nu_2}$ with $\boldsymbol{\nu_2}$ the unitary vector orthogonal to $\boldsymbol{\tau_2}$. The angle of the first trailer is then $\theta_1 = \theta_2 + atan(\kappa_2)$. We then deduce the path $\gamma_1$ followed by the first trailer. Parametrizing $\gamma_1$ 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 $\gamma$ followed by the robot when the second trailer follows $\gamma_2$. The only required condition is the existence of $\frac{d^2}{ds_2^2}\kappa_2$; moreover the relative angles $\varphi_1$ and $\varphi_2$ should belong to $]-\frac{\pi}{2},\frac{\pi}{2}[$ (see [26] for details).

Two configurations $X^{start}$ and $X^{goal}$ being given, one computes geometrically the values of $\kappa^{start}$, $\kappa_1^{start}$, $\kappa_2^{start}$, $\kappa^{goal}$, $\kappa_1^{goal}$ and $\kappa_2^{goal}$; each of them being a function of $\kappa_2$ and its derivative, it is straightforward to compute $\gamma_2$ satisfying such initial and final conditions (e.g., by using polynomial curves).

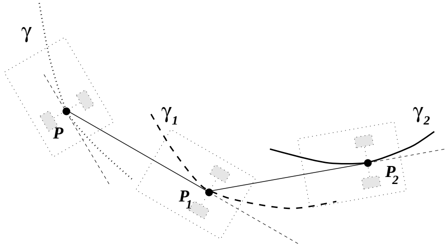


**Fig. 5.** Geometric construction of $P_1$ (resp. $P$) path from $P_2$ (resp. $P_1$) path.

*Remark:* Because the curvature of $\gamma_2$ 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 [63]. We will see below how to overcome this difficulty.

*Flatness conditions:* In cite [64], 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 $\Delta^k$ 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 2 (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 2 (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 2 (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 [63]). 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 [48,43] 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 Souères–Boissonnat's Chapter). For general systems, the only possibility is to call on numerical methods.

We sketch here the method developed by Fernandez, Li and Gurvitz in [24].

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[10] $\hat{u} = \sum_{k=1}^{N} \alpha_k e_k$, $e_k \in \{e^{i\frac{2p\pi}{T}},\ p \in \mathbf{Z}\}$. The choice of the reals $\alpha_k$ 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 [24]) that the solutions of the new finite-dimensional problem converge to the solutions of the original problem as $N$ and $\gamma$ go to infinity.

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 6 from [71] shows three examples of near-optimal paths computed from this method for a mobile robot with two trailers [49].

## 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 $(x, y, \theta)$ the robot first

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

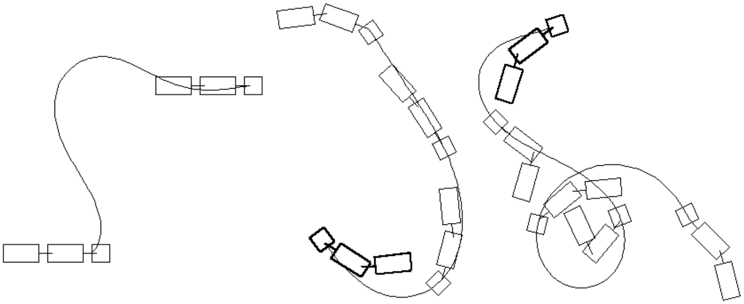


**Fig. 6.** Three examples of near-optimal paths.

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 17): the allowed mobile robot orientations $\theta$ 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 $\epsilon$.

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:*

$$\forall \epsilon > 0, \forall X^1 \in \mathcal{CS}, \exists \eta > 0, \forall X^2 \in \mathcal{CS}, \tag{10}$$

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

By using a steering method that verifies the weak topological property, it is possible to approximate any collision-free path $\gamma_{free}$. Nevertheless, this property is not sufficient from a *computational* point of view. Indeed, it is *local*: the real number $\eta$ depends on $X^1$. Situations as shown in Figure 7 may appear: let us consider a sequence of configurations $X^i$ converging to the critical point $X^c$, and such that $\lim_{X^i \rightarrow X^c} \eta(X^i) = 0$; 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.

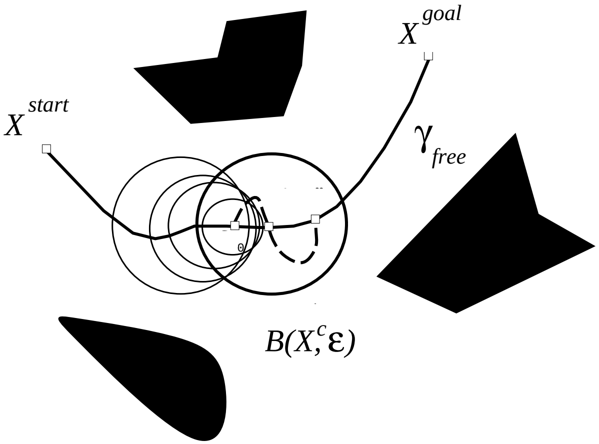


**Fig. 7.** Weak topological property

**Definition 3.** Steer *verifies the topological property iff:*

$$\forall \epsilon > 0, \exists \eta > 0, \forall (X^1, X^2) \in (\mathcal{CS})^2, \tag{11}$$

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

In this definition $\eta$ does not depend on any configuration (Figure 8). This is a global property that not only accounts for small-time controllability but also holds uniformly everywhere.

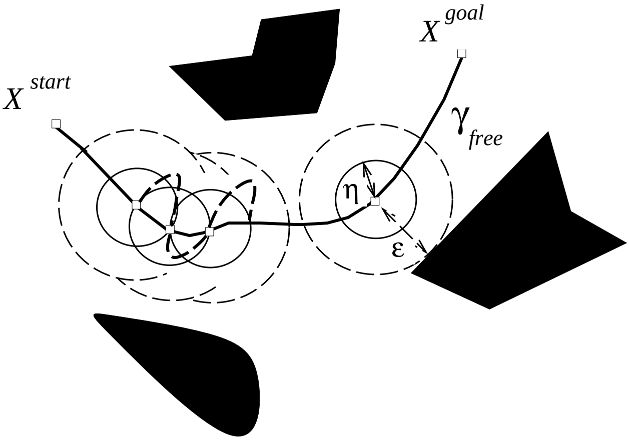


**Fig. 8.** Topological property

*Remark 1:* Proving that a given steering method verifies the topological property is not an easy task. The following sufficient condition appears in [71,74]. Let us equip $\mathcal{P}$ with a metric $d_{\mathcal{P}}$ between paths: $\Gamma_1$ and $\Gamma_2$ 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 $\mathsf{Steer}$ continuous w.r.t. to the topology induced by $d_{\mathcal{P}}$. $\mathsf{Steer}$ is uniformly continuous on any compact set $\mathcal{K}$ included in $\mathcal{CS}^2$, i.e.,

$$\begin{aligned}
&\forall\epsilon > 0\ ,\ \exists\eta > 0,\ \ \forall\ \{(X^1,X^2),(Y^1,Y^2)\} \in \mathcal{K}\\
&\qquad\qquad d_{CS}((X^1,Y^1) < \eta \ \text{ and }\ d_{CS}((X^2,Y^2) < \eta\\
&\qquad\qquad\qquad\qquad \Longrightarrow\ \ d_{\mathcal{P}}(\mathsf{Steer}(X^1,X^2),\mathsf{Steer}(Y^1,Y^2)) < \epsilon
\end{aligned}$$

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:

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

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

Therefore a sufficient condition for a steering method $\mathsf{Steer}$ to verify the topological property is that (1) $\mathsf{Steer}$ 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 [84]: 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 [87], 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.

**Optimal paths** Let us denote by $\mathsf{Steer}_{opt}$ the steering method using optimal control. $\mathsf{Steer}_{opt}$ 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 [92], or singular [14], or Carnot-Caratheory [56], or sub-Riemannian [80] 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 $\mathrm{d}_{\mathcal{CS}}$. This means that for any nonholonomic ball $B_{nh}(X, r)$ with radius $r$, there are two real numbers $\epsilon$ and $\eta$ 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 $\mathsf{Steer}_{opt}$ to a compact domain of the configuration space[11].

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 Bellaïche–Jean–Risler's chapter).

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 1).

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

**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 4). 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 $\mathsf{Steer}_{sin}^{a_1}$. For a fixed value of $a_1$, $\mathsf{Steer}_{sin}^{a_1}$ does not verify the topological property. Indeed, for any configuration $Z$, the path $\mathsf{Steer}_{sin}^{a_1}(Z,Z)$ is not reduced to a point[12].

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:

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

The proof appears in [73]. 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:

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

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

Then, by tuning $a_1$, it is *a priori* possible to design a steering method $\mathsf{Steer}_{sin}$ 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 $\mathsf{Steer}_{sin}$. 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 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.

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

**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 $\gamma_2$ (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 $(\ .\ ,\ .\ , \theta_2 \pm \frac{\pi}{2}, \ .\ ,\ .\ )$.

[74] takes advantage of the flatness property of a mobile robot with one trailer to design a steering method verifying the topological property. [41] 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 $\Gamma$ is an admissible path in the configuration space, we will denote by $\gamma_2$ the curve in $\mathbf{R}^2$ followed by $P_2$. Among all the admissible paths containing a configuration $X$, there exists exactly one path $\Gamma$ such that $\frac{d}{ds_2}\kappa_2$ remains constant everywhere: the corresponding curve $\gamma_2$ is a clothoid[13].

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 $X^{goal}$, $\alpha$ 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 $\gamma$ 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 $\mathsf{Steer}^*_{flat}$) 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 [41] we prove that, for any $\epsilon > 0$, if we choose a configuration $X$ within a "cone" $\mathcal{C}_{start,\epsilon}$ centered around $\Gamma_{start}$ with vertex $X^{start}$ (see Figure 9), 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

[13] If we consider only one trailer, we impose $\kappa_1$ to remain constant; in this case the trailer follows a circle with radius $\ell.\mathrm{cotan}\varphi_1$

$(x_2, y_2)$ (respectively $\theta_2$) 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 $\mathcal{C}_{start,\epsilon}$.

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

- If $X^{goal}$ belongs to $\mathcal{C}_{start,\epsilon}$, $\mathsf{Steer}_{flat}(X^{start}, X^{goal}) = \mathsf{Steer}^*_{flat}(X^{start}, X^{goal})$
- otherwise, we choose a point $X^{cusp}$ on $\gamma_{2,goal}$ within $\mathcal{C}_{start,\epsilon}$ 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 9).

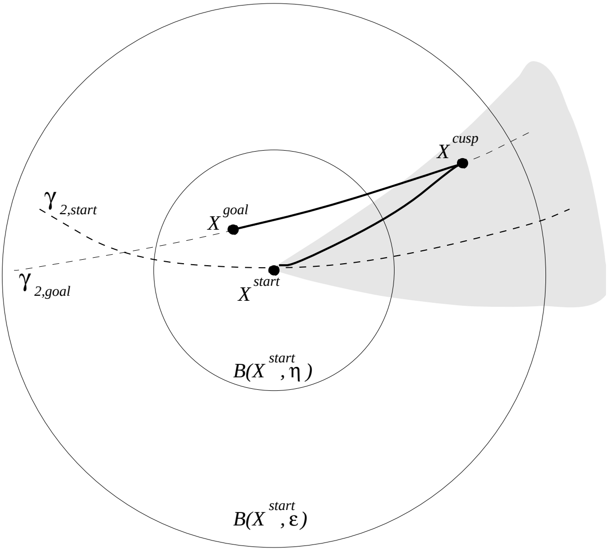


**Fig. 9.** Topological property of $\mathsf{Steer}_{flat}$.

Figure 10 shows an example of the path generated by $\mathsf{Steer}_{flat}$.

Figures 11 shows paths computed by $\mathsf{Steer}_{flat}$ for a mobile robot with one trailer (the curve is the path followed by the robot).

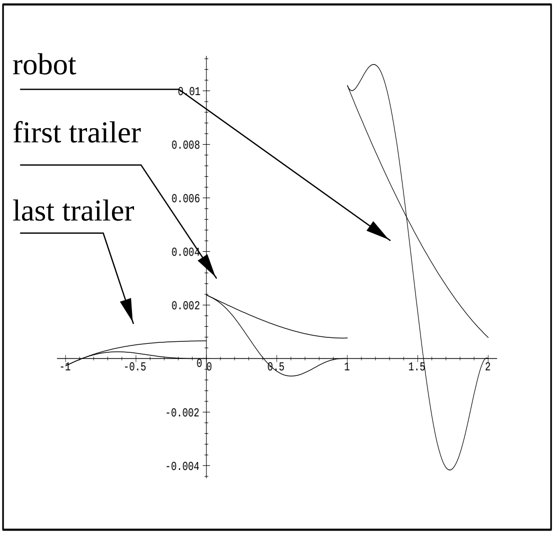


**Fig. 10.** An admissible path for a mobile robot with 2 trailers

### 5.3 Approximating holonomic paths: a two step approach

**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.

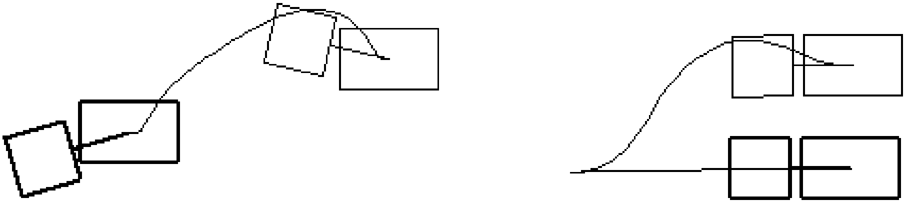


**Fig. 11.** Admissible paths computed by $\mathsf{Steer}_{flat}$ for a mobile robot with one trailer

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 3.1, 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[14].

*Geometric planner:* There are no general and practical algorithm solving the classical "piano mover" problem with completeness property[15]. Numerous techniques are available to address dedicated problems [42]. 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 (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.

[14] 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.

[15] General algorithms have been provided within the framework of real algebraic geometry [69,18]; 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.

**Application to mobile robots (without trailer)** The seminal ideas of the algorithm above have been introduced in [48]. 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 $\mathsf{Steer}_{opt}$: it consists in computing the shortest length admissible paths for a car-like robot as characterized in [65]. 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 [43] 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 12 shows an example of a solution from a software developed for the mobile robots Hilare at LAAS.

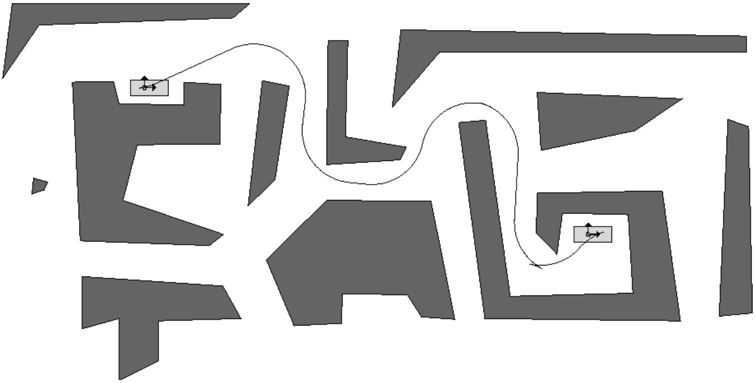


**Fig. 12.** A planned path for a car-like robot: the workspace is modeled by a grid of 250 × 150 pixels; the total running of the algorithm is 2 seconds on an Indy Silicon Graphics.

A clever idea appears in [55]. 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 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 above and Souères–Boissonnat's Chapter) provide an analytical way to compute the shortest path distance to a polygonal obstacle for a point car-like robot [91]. 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 13 from [38] 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.

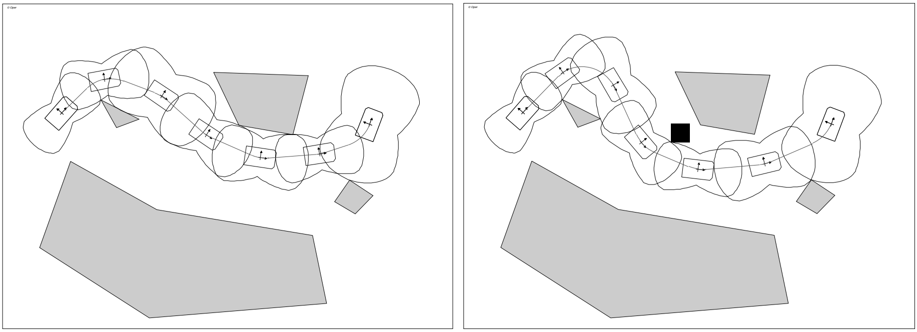


**Fig. 13.** A planned path is updated in real-time when an unexpected moving obstacle occurs.

**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 $\mathsf{Steer}_{opt}$, $\mathsf{Steer}_{sin}$ (for $n = 1$ and $n = 2$) and $\mathsf{Steer}_{flat}$ (for $n = 1$) [71].

To compute a collision-free path we use the algorithm RPP, the random 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[16]; 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 [71], it appears that the the algorithm based on $\mathsf{Steer}_{sin}$ is much faster than the algorithm based on $\mathsf{Steer}_{opt}$ (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 $\mathsf{Steer}_{flat}$ and $\mathsf{Steer}_{sin}$; nevertheless the smoothness of the final path is better with $\mathsf{Steer}_{flat}$ than with $\mathsf{Steer}_{sin}$.

In the examples on Figures 14 and 15 the workspace is modeled by a grid of 600 × 470 pixels.

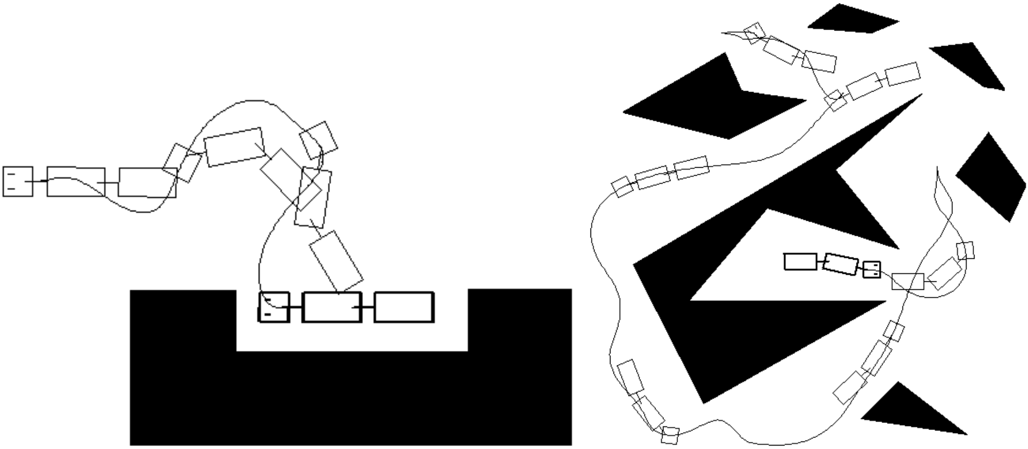


**Fig. 14.** Solutions using $\mathsf{Steer}_{sin}$: the total computation time is 30 seconds (left) and 114 seconds (right) on a Sun-Sparc-20.

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

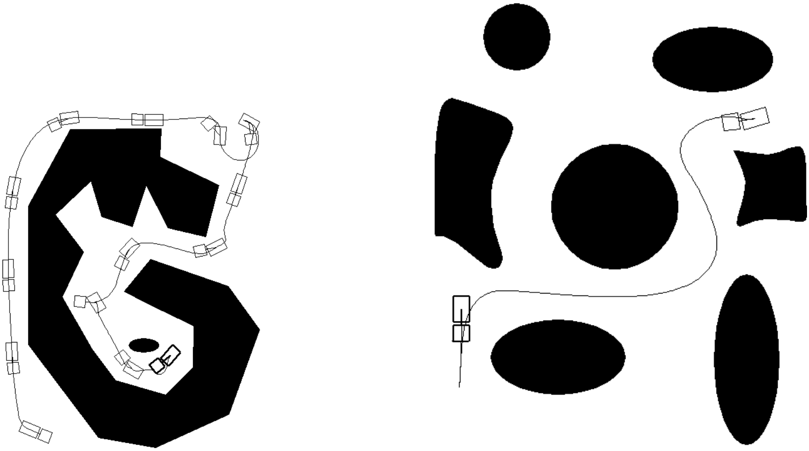


**Fig. 15.** Solutions using $\mathsf{Steer}_{flat}$: the total computation time is 21 seconds (left) and 6 seconds (right) on a Sun-Sparc-20.

### 5.4 Probabilistic approaches

Svestka–Overmars' chapter reviews recent results provided by applying a new general paradigm in motion planning. This is a probabilistic approach consisting in two phases:

- 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 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 $\mathsf{Steer}_{sin}$ [70]. 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* [12]. 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 $\mathsf{Steer}_{sin}$ as local path planner. Because $\mathsf{Steer}_{sin}$ 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. [70] 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 [70], 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 $\mathsf{Steer}_{sin}$ 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 $\mathsf{Steer}_{sin}$ 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 16: 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.

### 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 [42] 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 $\mathsf{Steer}_{opt}$. 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 13 is 7 in both cases. This definition allows to link the complexity of nonholonomic path planning with the clearance of the free-space.

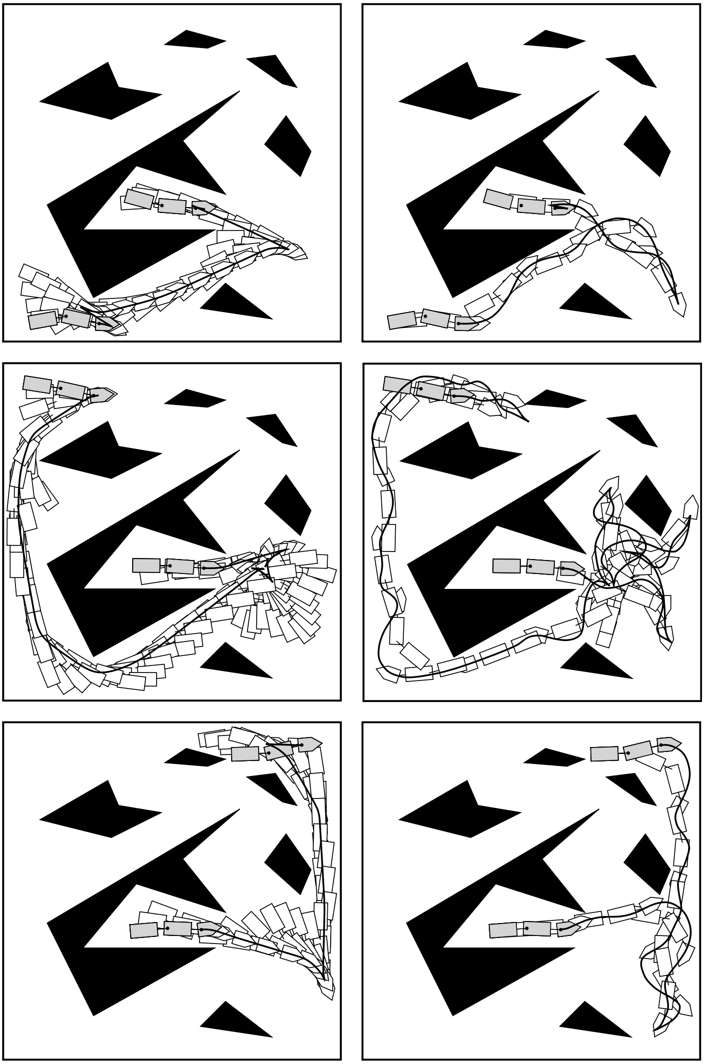


**Fig. 16.** Examples of solutions computed by the multi-level approach (the left column shows the first "semi-holonomic" paths)

Let us consider the classical parking task problem illustrated in Figure 17 for a car-like robot. The solutions have been computed by the algorithm presented in Section 15. The steering method to approximate the holonomic path is $\mathsf{Steer}_{opt}$ 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 1 (top). Let us consider a configuration $X = (x, y, \theta)$ near the origin $O$. It has been proved in [48] that:

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

As a consequence, the number of balls required to cover the "corridor" where the car has to be parked varies as $\epsilon^{-2}$ with $\epsilon$ 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 $\Omega(\epsilon^{-2})$.

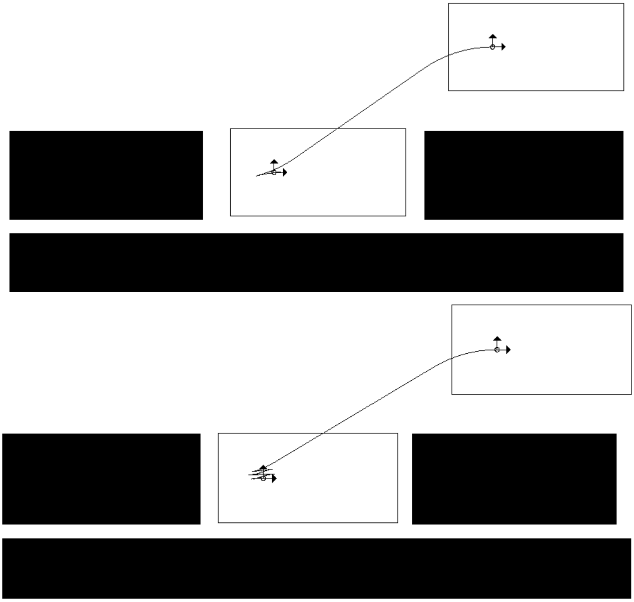


**Fig. 17.** The number of maneuvers varies as the inverse of the square of the free space.

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 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 [9,29,92,56] or Bellaïche–Jean–Risler's chapter): 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 $d$.

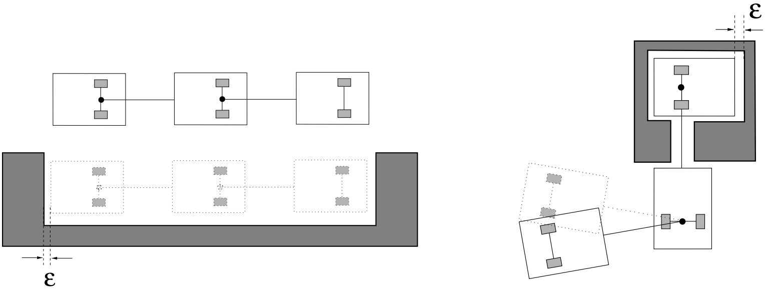


**Fig. 18.** The complexity of admissible paths for a mobile robot with $n$ trailers are respectively $\Omega(\epsilon^{-n-2})$ (case on the left side) and $\Omega(\epsilon^{-Fib(n+3)})$ (case on the right side).

Figure 18 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 from 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 $\frac{\pi}{2}$ [54,36] ($Fib(n+3)$ 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 18 and generalized to $n$ trailers are respectively $\Omega(\epsilon^{-n-2})$ (simply exponential in $n$) and $\Omega(\epsilon^{-Fib(n+3)})$ (doubly exponential in $n$).

## 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. 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 $\delta t$. 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 $\delta t$. 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 $\delta t$ 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 [23] 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 [22], 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., [45,89,94]), they are only few results addressing the general problem. All of them assume that the robot is reduced to a point. In [27], 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 [34] 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) [2,13].

Nonholonomic path planning for Dubins' car then remains a difficult and open problem[17].

*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[18].

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 [32], 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 [85]: 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

[17] Notice that Barraquand and Latombe's algorithm [6] may be applied to provide an approximated solution of the problem.

[18] We refer the reader to Svestka–Overmars' chapter for a more detailed overview on this topic.

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 Conclusions

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 2 (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 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.

## Annex: Sinusoidal inputs and obstacle avoidance (comments on the tuning of $a_1$)

As we have seen in Section 11, we do not dispose of a unique expression of $\mathsf{Steer}_{sin}$ 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 a general nonholonomic path planning scheme.

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:

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

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 [72,71]. 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 7.1.** *For any path computed by* $\mathsf{Steer}_{sin}^{a_1}$*, for any* $t \in [0, T]$ *:*

$$\begin{array}{ll} |\delta_1(t)| & \leq |a_0 T| + |a_1 T| = \varDelta_1 \\ |\delta_2(t)| & \leq \sum |b_i T| = \varDelta_2 \\ |\delta_{k+1}(t)| & \leq |z_k^0|\varDelta_1 + \ldots + |z_3^0|\varDelta_1^{k-2} + (|z_2^0| + \varDelta_2)\varDelta_1^{k-1} \quad \mathit{withk} > 2 \end{array} \tag{12}$$

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

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

By setting $\varDelta_1 = |a_0 T| + |a_1 T|$ we have the intermediate result that for all $t$, $\int_0^t |\dot{\delta}_1(\tau)| d\tau \leq \varDelta_1$. The same reasoning holds to prove that $|\delta_2(t)| \leq \sum |b_i T|$.

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 $\varDelta_k$ on $|\delta_k(t)|$ being given, we get:

$$|\delta_{k+1}(t)| \leq \varDelta_k \int_0^t |\dot{z}_1(\tau)|\, d\tau + |z_k^0| \int_0^t |\dot{z}_1(\tau)|\, d\tau \leq (\varDelta_k + |z_k^0|)\varDelta_1$$

Then

$$\varDelta_{k+1} \leq (\varDelta_k + |z_k^0|)\varDelta_1$$

And by recurrence:

$$|\delta_{k+1}(t)| \leq |z_k^0|\varDelta_1 + \ldots + |z_3^0|\varDelta_1^{k-2} + (|z_2^0| + \varDelta_2)\varDelta_1^{k-1} \quad \square$$

Given a start configuration $Z^{start}$, we first fix the value of $a_1$ and two other parameters $\varDelta_1^{min}$ and $\varDelta_2^{min}$ to some arbitrary values (see [71] 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 $\varDelta_1$ and $\varDelta_2$ (the detailed proof of this statement appears in [71,74]). We continue to bring the goal closer to the initial configuration until a collision-free path is found or until $\varDelta_1 \leq \varDelta_1^{min}$ and $\varDelta_2 \leq \varDelta_2^{min}$. In the second case, we substitute $a_1$, $\varDelta_1^{min}$ and $\varDelta_2^{min}$ respectively by $k.a_1$, $k.\varDelta_1^{min}$ and $k.\varDelta_2^{min}$, 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 (12) 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.

## 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," *International 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 International 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. 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.
13. 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.
14. R.W. Brockett, "Control theory and singular Riemannian geometry," *New Directions in Applied Mathematics*, Springer-Verlag, 1981.
15. X.N. Bui, P. Souères, J.D. Boissonnat and J.P. Laumond, "The shortest path synthesis for nonholonomic robots moving fordwards," *IEEE Int. Conf. on Robotics and Automation*, Atlanta, 1994.
16. L. Bushnell, D. Tilbury and S. Sastry, "Steering three-input nonholonomic systems: the fire-truck example," *International Journal of Robotics Research*, Vol. 14 (4), pp. 366–381, 1995.

17. 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.
18. J. Canny, *The Complexity of Robot Motion Planning*, MIT Press, 1988.
19. 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.
20. B. Donald, P. Xavier, J. Canny and J. Reif, "Kinodynamic motion planning," *J. of the ACM*, Vol. 40, pp. 1048–1066, 1993.
21. 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.
22. 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.
23. P. Ferbach, "A method of progressive constraints for nonholonomic motion planning," *IEEE Int. Conf. on Robotics and Automation*, pp. 2929–2955, Minneapolis, 1996.
24. 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.
25. 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.
26. 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.
27. S.J. Fortune and G.T. Wilfong, "Planning constrained motions," *ACM STOCS*, pp. 445–459, Chicago, 1988.
28. 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.
29. V. Y. Gershkovich, "Two-sided estimates of metrics generated by absolutely nonholonomic distributions on Riemannian manifolds," *Soviet Math. Dokl.*, Vol. 30 (2), 1984.
30. 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.
31. H. Hermes, A. Lundell and D. Sullivan, "Nilpotent bases for distributions and control systems," *J. of Differential Equations*, Vol. 55, pp. 385–400, 1984.
32. 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," *International Journal of Robotics Research*, Vol. 3, pp. 76–88, 1984.
33. G. Jacob, "Lyndon discretization and exact motion planning," *European Control Conference*, pp. 1507–1512, Grenoble, 1991.
34. P. Jacobs and J. Canny, " Planning smooth paths for mobile robots," *IEEE Int. Conf. on Robotics and Automation*, Scottsdale, 1989.

35. 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.
36. 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.
37. Y. Kanayama and N. Miyake, “Trajectory generation for mobile robots,” *Robotics Research*, Vol. 3, MIT Press, pp. 333–340, 1986.
38. 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.
39. G. Lafferriere and H.J. Sussmann, “A differential geometric approach to motion planning,” *Nonholonomic Motion Planning*, Zexiang Li and J.F. Canny Eds, The Kluwer International Series in Engineering and Computer Science 192, 1992.
40. F. Lamiraux and J.P. Laumond, “From paths to trajectories for multi-body mobile robots,” *Int. Symposium on Experimental Robotics*, Lecture Notes on Control and Information Science, Springer-Verlag, (to appear) 1997.
41. F. Lamiraux and J.P. Laumond, “Flatness and small-time controllability of multibody mobile robots: applications to motion planning,” *European Conference on Control*, Brussels, 1997.
42. J.C. Latombe, *Robot Motion Planning,* Kluwer Academic Publishers, 1991.
43. J.C. Latombe, “A Fast Path Planner for a Car-Like Indoor Mobile Robot,” *Ninth National Conference on Artificial Intelligence, AAAI*, pp. 659–665, Anaheim, 1991.
44. J.P. Laumond, “Feasible trajectories for mobile robots with kinematic and environment constraints,” *Intelligent Autonomous Systems*, L.O. Hertzberger, F.C.A. Groen Edts, North-Holland, pp. 346–354, 1987.
45. J.P. Laumond, “Finding collision-free smooth trajectories for a non-holonomic mobile robot,” *10th International Joint Conference on Artificial Intelligence*, pp. 1120–1123, Milano, 1987.
46. 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.
47. J.P. Laumond, “Controllability of a Multibody Mobile Robot,” *IEEE Transactions Robotics and Automation*, pp. 755–763, Vol. 9 (6), 1993.
48. 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.
49. 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.
50. J.P. Laumond and J.J. Risler, “Nonholonomic systems: controllability and complexity,” *Theoretical Computer Science*, Vol. 157, pp. 101–114, 1996.
51. 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.
52. Z. Li and J.F. Canny Eds, *Nonholonomic Motion Planning*, Kluwer Academic Publishers, 1992.

53. T. Lozano-Pérez, “Spatial planning: a configuration space approach,” *IEEE Trans. Computers*, Vol. 32 (2), 1983.
54. F. Luca and J.J. Risler, “The maximum of the degree of nonholonomy for the car with $n$ trailers,” in *IFAC Symp. on Robot Control*, pp. 165–170, Capri, 1994.
55. B. Mirtich and J. Canny, “Using skeletons for nonholonomic path planning among obstacles,” *IEEE Int. Conf. on Robotics and Automation*, Nice, 1992.
56. J. Mitchell, “On Carnot-Carathéodory metrics,” *J. Differential Geometry*, Vol. 21, pp. 35-45, 1985.
57. 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.
58. R.M. Murray and S. Sastry, “Steering nonholonomic systems using sinusoids,” *IEE Int. Conf. on Decision and Control*, pp. 2097–2101, 1990.
59. R.M. Murray, “Robotic Control and Nonholonomic Motion Planning,” PhD Thesis, Memorandum No. UCB/ERL M90/117, University of California, Berkeley, 1990.
60. 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.
61. N.J. Nilsson, “A mobile automaton: an application of artificial intelligence techniques,” *1st Int. Joint Conf. on Artificial Intelligence*, pp. 509–520, Washington, 1969.
62. C. O’Dunlaing, “Motion planning with inertial constraints,” *Algorithmica*, Vol. 2 (4), 1987.
63. P. Rouchon, M. Fliess, J. Lévine and P. Martin, “Flatness and motion planning: the car with $n$ trailers,” *European Control Conference*. pp. 1518–1522, 1993.
64. P. Rouchon, “Necessary condition and genericity of dynamic feedback linearization,” in *J. Math. Systems Estimation Control*, Vol. 4 (2), 1994.
65. 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.
66. 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.
67. 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.
68. J.J. Risler, “A bound for the degree of nonholonomy in the plane,” *Theoretical Computer Science*, Vol. 157, pp. 129–136, 1996.
69. 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.
70. S. Sekhavat, P. Švestka, J.P. Laumond and M. H. Overmars, “Multi-level 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.

71. S. Sekhavat, “Planification de mouvements sans collisions pour systèmes non holonomes,” PhD Thesis 1240, INPT, LAAS-CNRS, Toulouse, 1996.
72. 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.
73. 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).
74. S. Sekhavat, F. Lamiraux, J-P. Laumond, G. Bauzil and A. Ferrand, “Motion planning and control for Hilare pulling a trailer: experimental issues,” in *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, 1997.
75. 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.
76. P. Souères and J.P. Laumond, “Shortest path synthesis for a car-like robot,” *IEEE Trans. on Auromatic Control*, Vol. 41 (5), pp. 672–688, 1996.
77. E.D. Sontag, “Controllability is harder to decide than accessibility,” *SIAM J. Control and Optimization*, Vol. 26 (5), pp. 1106–1118, 1988.
78. 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.
79. S. Sternberg, *Lectures on Differential Geometry*, Chelsea Pub., 1983.
80. R. S. Strichartz, “Sub-Riemannian geometry,” *Journal of Differential Geometry*, Vol. 24, pp. 221–263, 1986.
81. R. S. Strichartz, “The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations,” *Journal of Functional Analysis*, Vol. 72, pp. 320–345, 1987.
82. H.J. Sussmann and V. Jurdjevic, “ Controllability of nonlinear systems,” *J. of Differential Equations*, 12, pp. 95–116, 1972.
83. 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.
84. 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.
85. 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.
86. 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.
87. 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.
88. A. Thompson, “The navigation system of the JPL robot,” *5th Int. Joint Conf. on Artificial Intelligence*, pp. 749–757, Cambridge, 1977.

89. 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.
90. V.S. Varadarajan, *Lie Groups, Lie Algebra and their Representations*, Springer-Verlag, 1984.
91. 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.
92. A.M. Vershik and V.Ya. Gershkovich, “Nonholonomic problems and the theory of distributions,” *Acta Applicandae Mathematicae*, Vol. 12, pp. 181–209, 1988.
93. X. Viennot, *Algèbres de Lie libres et monoïdes libres.* Lecture Notes in Mathematics, 691, Springer Verlag, 1978.
94. G.T. Wilfong, “ Motion planning for an autonomous vehicle,” *IEEE Int. Conf. on Robotics and Automation*, pp. 529–533, 1988.