A Motion Planner for Nonholonomic Mobile Robots

Jean-Paul Laumond, Paul E. Jacobs, *Member, IEEE,* Michel Taïx, and Richard M. Murray, *Member, IEEE*

***Abstract*—This paper considers the problem of motion planning for a car-like robot (i.e., a mobile robot with a nonholonomic constraint whose turning radius is lower-bounded). We present a fast and exact planner for our mobile robot model, based upon recursive subdivision of a collision-free path generated by a lower-level geometric planner that ignores the motion constraints. The resultant trajectory is optimized to give a path that is of near-minimal length in its homotopy class. Our claims of high speed are supported by experimental results for implementations that assume a robot moving amid polygonal obstacles.**

The completeness and the complexity of the algorithm are proven using an appropriate metric in the configuration space $\mathbf{R}^2 \times S^1$ of the robot. This metric is defined by using the length of the shortest paths in the absence of obstacles as the distance between two configurations. We prove that the new induced topology and the classical one are the same. Although we concentrate upon the car-like robot, the generalization of these techniques leads to new theoretical issues involving sub-Riemannian geometry and to practical results for nonholonomic motion planning.

I. Introduction

IN THIS PAPER we present a motion planner for nonholonomic mobile robots. The goal of this work is two-fold. From a practical viewpoint, we have sought a planner that is exact and yet requires very little computation time. Experimental results highlight its high speed. On the theoretical level, we wish to link the geometric formulation of motion planning to the dynamical systems formulation of differential geometric control theory. This linkage is a recent development in the robotics community. One goal of this paper is to illustrate its utility using the car-like system as an example.

This study has been done in the context of the Hilare mobile robot project (see Fig. 1) being pursued at LAAS/CNRS [11], [16].

A. The Problem of Motion Planning with Nonholonomic Constraints

Motion planning is one of the better-formulated problems in robotics. Its geometric formulation is the classical "piano-mover problem," in which the problem of the motion of rigid bodies amid obstacles in the 3-dimensional Euclidean space is translated into the problem of the motion of a point in some space, called the configuration space. With this formulation, the existence of a collision-free trajectory for the body is characterized by the existence of a connected component in the admissible (i.e., collision-free) configuration space. Many papers have proposed general, specific, exact, approximate, efficient, or inefficient methods in order to represent and explore this admissible configuration space (see [25] for a synthesis of these approaches).

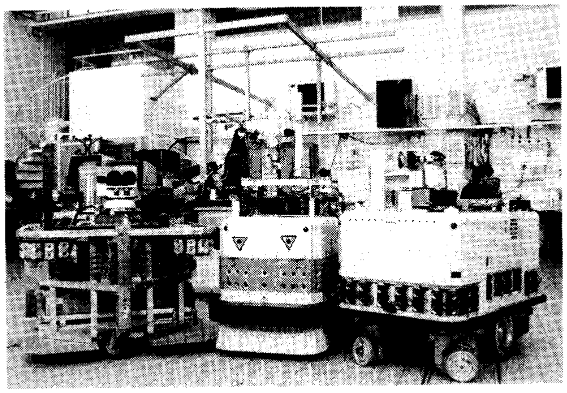


Fig. 1. The Hilare family.

However, there are many cases in which this general formulation of motion planning does not hold. This paper deals with the problem of planning constrained motions where the constraints are nonholonomic in nature. More specifically, it appears that the numerous classical methods work only when the robotic system is holonomic and not when it has some nonholonomic constraint between its configuration parameters. As an example, the kinematic constraints imposed by the fact that there are joints connecting the links of a robot manipulator are holonomic. It is for this reason that we can construct a configuration space of joint angles that completely characterizes the location of the manipulator in space. The allowable velocities of a point in this space are precisely the elements of the tangent space of the manifold of configurations. In contrast, a nonholonomic constraint is expressed as a non-integrable equation involving the derivatives of the configuration parameters. Since this equation is not integrable, there are constraints in the tangent space at each configuration (that is, on the allowable velocities) and these cannot be eliminated by defining a more restricted configuration space manifold in which the points can then move in any direction. The main consequence of a nonholonomic constraint is that an arbitrary path in the admissible configuration space does not necessarily correspond to a feasible trajectory for the robot.

Manuscript received May 20, 1991; revised September 8, 1992. This work was supported in part by the European Esprit 3 Project PROMotion 6546, Groupement Robots d'Intervention sur Site Planétaire (RISP), and the David and Lucile Packard Foundation. The cooperative research program between LAAS and ERL was supported by NSF 87-87-19298.

J.-P. Laumond and M. Taïx are with LAAS/CNRS, 31077 Toulouse, France.

P. E. Jacobs was with LAAS/CNRS, 31077 Toulouse, France, on leave from Qualcomm, Inc., San Diego, CA 92121 USA.

R. M. Murray was with the University of California at Berkeley, Berkeley, CA 94720 USA. He is now with the Department of Mechanical Engineering, the California Institute of Technology, Pasadena, CA 91125 USA.

IEEE Log Number 9209587.

1042-296X/94$04.00 © 1994 IEEE

Therefore, the existence of a collision-free trajectory is not *a priori* characterized by the existence of a connected component in the admissible configuration space.

B. The Differential Geometric Control Theory Perspective

Motion planning with nonholonomic constraints was brought into the domain of robotics in [27]. It has attracted a significant amount of interest during the last few years. The most important results have been obtained by attacking the problem with tools from differential geometric control theory [5], [24], [30], [34], [37]. An overview of the tools linking motion planning together with control theory can be found in [32] and in Murray's thesis [39].[1] We summarize these results.

In the presence of a relationship between the robot's configuration variables and their derivatives, two questions arise:

1) Is this relationship holonomic? (i.e., does it reduce the dimension of the configuration space?)

2) If not, does it reduce the accessible configuration space?

In the case of r constraints corresponding to r equations linear in the n derivatives of the parameters, these equations determine what is called an $(n - r)$-distribution Δ on the manifold of configurations. Let us recall that the Lie bracket of two vector fields X and Y is defined as $[X, Y] = \partial X.Y - \partial Y.X$. The equations are integrable (or holonomic) if and only if Δ is closed under the Lie bracket operation (Frobenius' theorem, see for instance [48]).

From a control theory perspective, a control is a function that allows one to choose the system state velocity at a given time by a weighting of smooth vector fields. The control Lie algebra associated with Δ, denoted by $LA(\Delta)$, is the smallest distribution that contains Δ and is closed under the Lie bracket operation. The answer to the second question is then given by the non-linear system controllability theorem (see for instance [17], [35], [52]), which is known as the Lie algebra rank condition (LARC): if the rank of $LA(\Delta)$ is full at a given configuration c, then there exists a neighborhood of c, all of whose points are reachable by the system from c. In this case, the system is said to be *locally controllable* [53]. LARC is a *local* condition. If the rank condition holds everywhere in the configuration space, then we will say that the system is *controllable*. From the motion planning point of view, the existence of a collision-free trajectory for a controllable system is characterized by the existence of a connected component in the *free* (i.e., admissible and without contact) configuration space.

Unfortunately, this result answers the question of the existence of a feasible trajectory, but does not solve the complete problem of efficiently producing a trajectory. Nevertheless, as a consequence of the controllability theorem, we can hope that the search for a trajectory for a controllable nonholonomic system can be steered by a solution of the associated unconstrained one. This simple idea is the basis of our algorithm.

C. Related Work

The control theory point of view of motion planning for nonholonomic systems has been attacked without considering the presence of obstacles. Murray and Sastry in [38], [39] explored the use of sinusoidal controls in steering some class of nonholonomic systems: the class of systems that can be converted into a so-called "chained" form. Because of the special form, there exist simple sinusoidal iterated controls that generate motions steering a given coordinate to the goal while leaving the previous ones unchanged.

At the same time, Lafferiere and Sussmann [24] proposed a general approach to plan complete trajectories for controllable systems without drift. Their strategy generates piece-wise constant controls. An initial (not necessarily feasible) path is first assumed to be given. This path is then expressed locally on some special basis (P. Hall basis) of the tangent space. The coefficients computed from the linear combinations of the tangent vectors define controls that produce a feasible path approximating the initial one. In [18], Jacob gives an account of this strategy by using another basis (Lyndon basis) that reduces the number of pieces of the solution.

The obstacle avoidance problem for general systems has been attacked by Sussmann and Liu in a clever (and difficult) paper [54]. The authors give an algorithm for constructing a sequence of admissible trajectories that converge *uniformly* to any given path. This guarantees that one can choose an admissible trajectory that is *arbitrarily close* to a given collision-free path. Such a method has been applied to planning paths for a four-body mobile robot (i.e., a mobile robot with three trailers) [57]. As in [38], the method uses sinusoidal inputs.

Nevertheless, as pointed out by the authors, the main difficulty is determining the "size" of the free space allowed around the steering solution. The answer demands a study of the topologies that can be induced on the configuration space by different metrics. This study is developed in the current paper for the case of a car-like robot.

D. Overview of the Results

Our nonholonomic motion planner is based on the following scheme. First of all, a collision-free path is generated ignoring the nonholonomic and curvature constraints. Such a path can be generated by any geometric motion planner that solves the piano movers problem, giving a path that is strictly collision-free. In order to transform this path into one that is feasible with respect to the nonholonomic and curvature constraints, we make use of an exact characterization of the minimal-length constrained path connecting any two configurations, in the absence of obstacles. This characterization has been produced by Reeds and Shepp in [43]. In order to account for the obstacles, our algorithm iteratively computes a set of subgoals along the initial geometric path. These subgoals are chosen so that, even with the motion constraints, the minimal-length constrained path connecting successive pairs

[1] Appendix A gives an overview of other results related to nonholonomic motion planning: some of them do not extensively use the tools from differential geometric control theory, but they are interesting nonetheless, from either a theoretical or practical point of view; others are related to holonomic systems exhibiting conservation laws. Finally, recall that the motion planning problem consists of finding *open* -loop controls that steer the system from an initial configuration toward a goal. The problem of finding *closed*-loop controls is another difficult one (see [2], [9], [44], [61] for instance).

of configurations is collision-free. The complete trajectory formed by concatenation of the paths between subgoals is then collision-free *and* respects the nonholonomic and curvature constraints. Because the resultant feasible trajectory is highly dependent on the initial path that it approximates, it is then optimized to give a path that is of near-minimal length in its homotopy class. The algorithm we present is exact. No *a priori* assumptions need to be made in discretizing the environment. As an example, the algorithm can generate a collision-free feasible trajectory for parallel-parking the robot in an arbitrarily tight space, if one exists (Fig. 6).

The completeness and the complexity analysis of the algorithm are based on the study of the configuration-space topology defined from the metric induced by the length of the shortest feasible paths in the plane. After introducing the formal model of a system we term *the car-like robot* (Section II-A), we prove its controllability (Section II-B). Then we present the work by Reeds and Shepp on the shape of the shortest paths (Section II-C) and a topological property dealing with the metric induced by such paths (Section II-D). This topological property is linked with the sub-Riemannian geometric framework. It constitutes the main result for designing (Section III) and analyzing (Section IV) our algorithm. We then present experimental results obtained from two different implementations (Section V). Finally, we conclude with a generalization of the approach and we pinpoint the theoretical challenge involving optimal control theory and sub-Riemannian geometry. Appendices B and C present proofs of the controllability property and the topological property of Section II. A goal of the proof presentations is to illustrate the theoretical underpinnings of differential geometric control theory, mainly the Lie algebra rank condition (Appendix B), and the way to compute the shape of singular metrics in sub-Riemannian geometry (Appendix C).

It should be noted that the details of the operation of the algorithm (Section III) can be understood and used without any reference to the proofs of convergence and complexity.

II. Planning Paths for a Car- Like Robot

A. The Car-Like Robot

The system that we call the *car-like robot* is a simplified model of the constraints on the movement of a real car (Fig. 2). It takes into account both the nonholonomic constraint

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

that specifies the tangent direction along any feasible path for the robot and a bound on the curvature of the path.

It is perhaps more clear when we translate the kinematic constraints into a set of differential equations with the allowable controls explicitly specified.

From the driver's point of view, a car has two degrees of freedom: the accelerator and the steering wheel. We consider the midpoint of the rear wheels as the reference point. We assume that the distance between both rear and front axles is 1. We denote by v the speed of the front wheels of the car and by ϕ the angle between the front wheels and the main direction of the car.[2] Simple arguments show that the control system is the following:

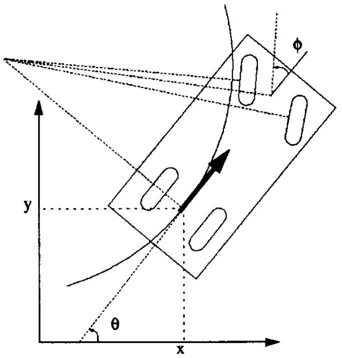


Fig. 2. A car-like robot.

$$\begin{pmatrix}\dot x\\ \dot y\\ \dot\theta\\ \dot v\\ \dot\phi\end{pmatrix} = \begin{pmatrix} v\cos\phi\cos\theta\\ v\cos\phi\sin\theta\\ v\sin\phi\\ 0\\ 0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} v_1 + \begin{pmatrix}0\\0\\0\\0\\1\end{pmatrix} v_2$$

Since we are concerned only with obstacle avoidance, neither the position of the front wheels nor the speed of the vehicle are relevant to the problem. This allows consideration of the following simplified system:

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

This system can be viewed as a nonlinear system whose controls are v and ϕ.

The constraint on the turning radius is expressed by $|\phi| \le \phi_0$ where ϕ_0 is a strictly positive real. Let us fix $\phi_0 = \frac{\pi}{4}$. Moreover we assume that the velocity of the car is upper-bounded by 1. The turning radius is then also upper-bounded by 1.

By putting $u_1 = v\cos\phi$ and $u_2 = v\sin\phi$, the system can be written in the following form.

$$\begin{pmatrix}\dot x\\ \dot y\\ \dot\theta\end{pmatrix} = \begin{pmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix} \tag{2}$$

with

$$|u_2(t)| \le |u_1(t)| \le 1 \tag{3}$$

The constraint on the controls must hold for all t in the domain of definition. For ease of reference, we write the vector fields separately:

$$X_1 = \begin{pmatrix}\cos\theta\\ \sin\theta\\ 0\end{pmatrix} \qquad X_2 = \begin{pmatrix}0\\0\\1\end{pmatrix} \tag{4}$$

[2] More precisely, the front wheels are not exactly parallel (see Fig. 2); we keep the average of their angles as the turn angle.

B. Controllability of the Car-Like Robot

In spite of the constraints on the controls given by inequalities (3), the proof of controllability is a straight-forward application of the nonlinear local controllability theorem that states that for the system to be controllable, the LARC condition holds. It suffices to consider the two constant admissible controls

$$\begin{pmatrix}1\\0\end{pmatrix} \text{ and } \begin{pmatrix}1\\1\end{pmatrix} \tag{5}$$

that respect the curvature bounds. The first control corresponds to a straight-line motion (corresponding to the vector field X_1), while the second one produces an arc of circle of minimal radius (corresponding to the vector field $X_1 + X_2$). The coordinates of $[X_1, X_1 + X_2]$ are:

$$\left[\begin{pmatrix}\cos\theta\\ \sin\theta\\ 0\end{pmatrix}, \begin{pmatrix}\cos\theta\\ \sin\theta\\ 1\end{pmatrix}\right] = \begin{pmatrix}-\sin\theta\\ \cos\theta\\ 0\end{pmatrix}. \tag{6}$$

One verifies easily that the Lie algebra generated by $X_1, X_1 + X_2$ and $[X_1, X_1 + X_2]$ is 3-dimensional everywhere in the configuration space, and so the system is controllable. Thus, even with curvature bounds, this system can be placed into the framework of differential geometric control theory.

Property 1: The car-like system is controllable.

Appendix B gives two direct proofs of this result. The first is presented without any reference to Lie brackets of vector fields. The second appears as a reading of the first one from the differential geometry point of view. It illustrates the theoretical foundations of the Lie algebra rank condition.

C. Shortest Paths for a Car-Like Robot

The study of shortest paths in the absence of obstacles for a system similar to the car-like robot was performed by Dubins [12]. The linear velocity control u_1 is fixed equal to one. In this case, the system has a drift term and is no longer symmetric. Dubins proves that the shortest paths are curves of class C^1 composed of arcs of circle with radius 1 and straight line segments. It is interesting to note that the system is locally controllable. Nevertheless, it is possible to determine initial and final configurations that are arbitrarily close, but that require paths whose lengths do not converge to zero: the system is no longer small-time locally controllable (see Appendix B for details about this notion).

Reeds and Shepp [43] have extended the work of Dubins to the car-like system we consider here. Because the linear velocity control u_1 can take on both positive and negative values, they allow maneuvers, or cusps, along the path. The shortest paths are then piece-wise smooth and have bounded curvature where it is defined. It is clear that between cusps, the path must be of the form given by Dubins.

Essentially, their method is to show that any path with more than two cusps can be reduced to a path with at most two cusps, which is not longer and is possibly shorter. They then eliminate some of the allowable curves using a homotopy argument. Finally they obtain a *finite* family of curves containing a shortest path[3] One can then design an algorithm in order to compute a shortest path between any two configurations. Examples are shown in Fig. 3.

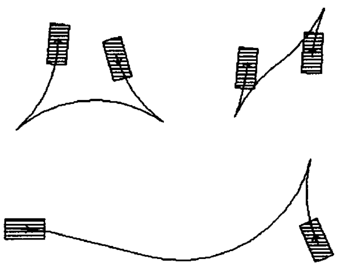


Fig. 3. Examples of shortest paths.

Notice that, as in Dubin's works, this characterization was done without obstacles.

New proofs of Reeds and Shepp's result have been obtained in the framework of the optimal control theory [8], [55] by using the maximum principle, a general theorem giving necessary conditions for trajectories to be optimal. Finally, it is possible to complete Reeds and Shepp's characterization of the shortest paths by providing the synthesis of all the shortest paths; i.e., by providing *necessary and sufficient* conditions of existence of the shortest paths for all the paths in the Reeds and Shepp's family, according to the start and goal configurations [47].

Remark 1: The model of Reeds and Shepp's system does not fit exactly with our model. Indeed, Reeds and Shepp assume the linear velocity to be constant and equal to 1, while our model just considers that the linear velocity is upper-bounded by 1. Nevertheless, Sussmann and Tang prove in [55] that the shortest paths for systems satisfying the inequalities (3) are the same as for Reeds and Shepp's problem.

D. The Shortest Path Metric and Sub-Riemannian Geometry

Because we have the exact form of the shortest paths for the car-like system, we can algorithmically compute the arclength in the plane of the shortest path connecting any two configurations. Furthermore, there is a one-to-one correspondence between the paths in the Euclidean plane $\mathbf{R}^2$ and the paths in the configuration space $\mathbf{R}^2 \times S^1$ that satisfy the nonholonomic constraint 1. Therefore, it is clear that such a distance is a metric on the configuration space. Let d_{RS} denote this metric.

The notion of a metric brings up the question of the nature of the induced topology. Not only that, but in the problem of motion planning for a mobile robot, we must also deal with obstacles that lie in the plane, and therefore with a standard Euclidean metric on the plane. By examining the distance between nearby configurations in the shortest-path metric, we can determine the reachable set in the presence of obstacles. We have then to compare both topologies. This is done by the following property that constitutes the main result of this paper:

[3] Shortest paths are not necessarily unique.

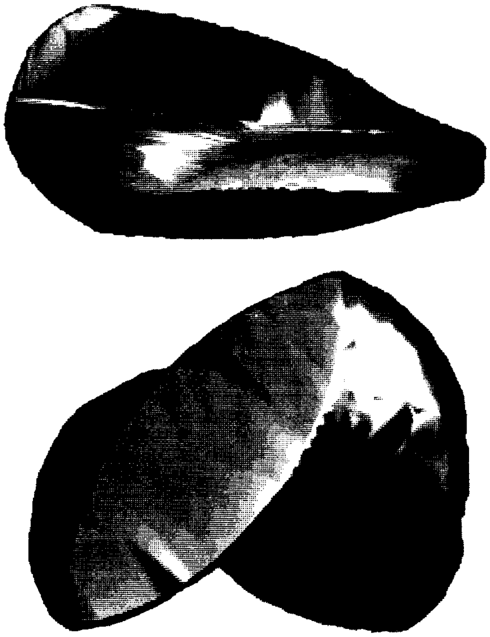


Fig. 4. The same ball from two points of view.

Property 2: For any point $c = (x, y, \theta)$ sufficiently near the origin $o = (0, 0, 0)$,

$$\tfrac{1}{3}(|x| + |y|^{1/2} + |\theta|) \le d_{RS}(c, o) \le 12(|x| + y|^{1/2} + |\theta|)$$

The proof of this property appears in Appendix C. The shape of the d_{RS} metric implies that the associated topology and the Euclidean one are the same.[4] More precisely:

Corollary: For each neighborhood (in the Euclidean topology) $\mathcal{N}(c)$ of a configuration c, there exists a neighborhood (in the Euclidean topology) $\mathcal{N}'(c)$ such that for any configuration $c' \in \mathcal{N}'(c)$ the path corresponding to the shortest path between c and c' is included in $\mathcal{N}(c)$.

The proof of the convergence of our algorithm below requires this precise corollary, while the complexity analysis is based on Property 2.

Fig. 4 shows a ball computed in the d_{RS} metric. When the radius of the ball increases the ball looks like a "cylinder." This phenomenon is not surprising: when c is far to the origin the associated shortest path contains a long straight line segment and its length is dominated by the length of this segment. In this case, $d_{RS}(c, o)$ is equivalent to $|x| + |y|$.

A general perspective to compare both topologies is to use the methods of sub-Riemannian geometry (see Strichartz's paper [50] for an introduction). Metrics can be defined by minimizing the length of all trajectories linking two given points. When the systems are nonholonomic, such metrics are said to be sub-Riemannian (or Carnot-Caratheodory, or singular). For the car-like system this metric is obtained by minimizing $\int(u_1^2 + u_2^2)^{1/2}dt$. The shortest path metric d_{RS} consists in minimizing the integral of the linear speed, i.e., $\int |u_1|dt$. This is not a sub-Riemannian metric. But we can prove [6] that both are equivalent. Finally, general arguments give the shape of the sub-Riemannian metric [15], [50] that would lead to state Property 2. The direct proof of Property 2 in Appendix C illustrates principles underpinning sub-Riemannian concepts and the ways to process them.

Notice that the connection between motion planning for nonholonomic systems and sub-Riemannian geometry is also investigated by Murray and Sastry [37] in the framework of grasping and manipulation by multifingered hands.

F. Computational Consequences

At this stage, let us summarize our results and let us link them to the motion planning problem.

The controllability of a car-like robot means that the existence of a collision-free feasible path for the system is characterized by the existence of a path included in an *open* set of the free configuration space for the associated holonomic system. In other words, any path for the holonomic system can be approximated by a feasible path. As we said in the introduction, the decision problem of motion planning is solved at this step.

The topological property (Corollary above) is more precise. It constitutes a local *constructive* proof of accessibility. It says that any path of the holonomic system (included in an open set of the admissible configuration space) can be discretized into a *finite* number of points such that, if one joins two consecutive points of the path by a Reeds-Shepp curve, one obtains a new path that constitutes a feasible collision-free path for the nonholonomic system.[5] The algorithm below applies this principle precisely. Finally Property 2 will be used in order to provide an upper bound of the number of subdividing points required by the method.

III. Operation of the Algorithm

The basic idea of the algorithm is to generate a path that is feasible with respect to the curvature and nonholonomic motion constraints by approximating a first path found by a lower-level geometric planner. In order to link intermediate configurations along the path generated by the geometric planner, we use the shortest path from the set of shortest feasible paths in the absence of obstacles, as given by Reeds and Shepp. Additional intermediate configurations are computed by recursively subdividing the geometric path whenever a collision is detected along a given feasible subpath. Hence, we start by trying to reach the goal directly from the initial configuration by the shortest feasible path. If this path intersects an obstacle, the configuration halfway along the solution path of the holonomic system is used as a subgoal. The problem is thus broken into two subproblems. If, at any point, the minimal length path between subgoals is not collision-free, then we recursively subdivide the holonomic path. When all of the necessary subdivisions are completed, the concatenation

[4] Indeed, this is the classical result that the Euclidean metric $\sqrt{x^2 + y^2 + \theta^2}$ and any metric of the shape $|x|^p + |y|^q + |\theta|^r$ induce the same topology.

[5] The number of points is easily proved to be finite by using the fact that the holonomic path is compact.

of all the feasible paths will be collision-free and respect to nonholonomic and curvature constraints.

Therefore, the algorithm that finds paths for a nonholonomic system consists of three steps:

1) Plan a collision-free path for the geometric system; i.e., without taking into account any nonholonomic constraint. If one does not exist, then no feasible path exists.
2) Perform subdivision on the path until all endpoints can be linked by a minimal-length path satisfying the nonholonomic and curvature constraints.
3) Perform an "optimization" routine to remove extra maneuvers and reduce the length of the path.

In the following, we detail each of the three steps. We then prove the convergence and the completeness of the algorithm in Section IV-A, while Section IV-B analyzes its complexity by providing an upper bound on the number of subdivisions of Step 2) above.

A. A Solution for the Holonomic System

The first part of our approach is to find a collision-free path for the robot without the nonholonomic and curvature constraints. This path characterizes the homotopy class in which the subdivision method will have to work. While we discuss a specific global motion planner in this work, we wish to make clear that this approach can be based upon any motion planner that can return a path lying wholly in the free configuration space. Experimental results will present two implementations based on two different geometric planners. The first one runs for a polygonal robot, the second one for a disk.

Over the past five years, results from computational geometry have allowed the efficient solution (i.e., by exact and complete techniques implementable with good practical and theoretical time complexity) of the problem of motion planning for a robot in a polygonal scene.

The general case of a non-convex polygon moving in a polygonal space has been attacked by Avnaim, Boissonnat, and Faverjon [3]. An exact boundary description of the free space is computed in $O(n^3 \log n)$ (where n is the number of edges of the environment). From this description a motion in contact can be found in $O(n^3)$.

In the same reference, authors present a cell decomposition of free space yielding a motion planner that produces paths without contact. The algorithm runs in $O(n^6\alpha(n))$ (where α is the inverse of the Ackermann function). This bound has been improved to $O(n^5 \log n)$ in [4].

The method we use produces semi-free paths (i.e., paths for which some portions are in contact), while the controllability result we have presented assumes that the trajectory of the holonomic system has to lie in an open set. We resolve this problem by assuming that the environment is in general disposition (i.e., the closure of the free configuration space is equal to the admissible-free and contact-configuration space; see [27] for details on the connectivity and the topology of the various configuration subspaces). With this hypothesis a trajectory in contact may always be displaced from the boundary in an open set of the free space (see [31] for details).

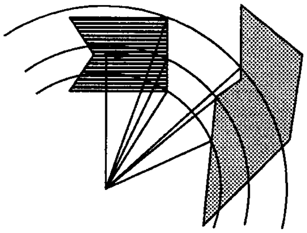


Fig. 5. Collision detection along a circular arc.

For the case of a circular robot we use a geometric planner based on the computation of the Voronoï diagram of a set of polygons, as described in [41]. The holonomic paths produced by such a planner are as far as possible from the obstacles.

B. Computation of a Feasible Path

1) General Scheme: At this point we assume that we have been given a path for the geometric system, which lies completely in the free configuration space. That is, at no point does the body of the robot touch an obstacle. Let us denote this path as $\gamma(s) \in \mathbf{R}^2 \times S^1$, where $s \in [0,1]$. Thus, $\gamma(0)$ is the initial configuration and $\gamma(1)$ is the final configuration. The problem is that the tangent to $\gamma(s)$, in general, does not satisfy the differential equation (1) or the control constraints (2) and (3). That is $\dot{\gamma}(s)$ may not be in the span of X_1 and X_2, or if it is, it may not satisfy the constraints on u_1 and u_2. The second step of the algorithm acts to generate a collision-free trajectory that does satisfy for these motion constraints by approximating the path $\gamma(s)$.

The operation of the algorithm is extremely simple. We begin by computing the minimal-length feasible trajectory (ignoring the obstacles) that connects the initial and final configurations. If this path is collision-free, we stop and return this trajectory. If there is a collision along the trajectory, we generate the configuration, $\gamma(\frac{1}{2})$, which is halfway between the initial and final configuration along the path provided by the geometric planner. We then generate the two minimal-length paths between the initial configuration and $\gamma(\frac{1}{2})$, and between $\gamma(\frac{1}{2})$ and the final configuration. If either of these two paths has a collision, we continue the subdivision $\gamma(s)$ in the appropriate subinterval of [0, 1], at each point generating and testing two minimal length paths for each original interval along $\gamma(s)$.

2) Collision Detection: Collision detection must be performed during the operation of the algorithm in order to determine when the process of subdivision has succeeded in a particular interval. It should be noted that the detection of collisions is performed for a certain pre-specified class of trajectories. In particular, for the car-like robot, the trajectories are composed solely of arcs of circles and straight-line segments. Such trajectories lend themselves to extremely rapid and exact collision detection, based solely upon the intersections of arcs of circles and line segments, as illustrated in Fig. 5. Furthermore, arctangents need not be computed, since the ordering of possible intersections by angle can be done by comparing the sines and cosines of the angles, which can be computed from the cross and dot product of the appropriate vectors.

In our current implementation, collision detection is performed by considering pairs of robot vertices and obstacle edges, and obstacle vertices and robot edges. For each portion of a trajectory, we perform $O(mn)$ operations, where m and n are the number of robot vertices and obstacle vertices, respectively, but it seems clear that with some preprocessing we could reduce this complexity.[6]

Finally, collision detection for the case of a disk is easy to implement efficiently.

3) Subdividing the Geometric Path: In our current implementation, the geometric path is returned as a list of intermediate configurations linked by a specified type of path. In order to avoid continually re-reading this list, the configurations are read into an array.[7] This array is then used as the basis for the subdivision algorithm. That is, the algorithm consists of a single recursive procedure that operates on the array of configurations, passing a sub-array to itself in the case that the first and last elements of the given array can not be linked by a collision-free minimal-length curve. In the case that the array consists of only two elements, we interpolate along the curves using our knowledge of the type of path linking the two configurations. In this way, the number of operations performed depends linearly upon the size of the smallest subinterval that must be generated. Precisely, it is $O(L/\Delta s + K)$, where L is the arclength of $\gamma(s)$, Δs is the size of the smallest sub-interval, and K is the number of subpaths originally used to specify $\gamma(s)$. We address the computation of Δs in Section IV-B.

C. Optimization of Feasible Trajectories

The trajectory that is generated by the action of the first two steps of the algorithm is both collision-free and feasible with respect to the nonholonomic and curvature constraints. However, its exact form is highly dependent on the initial trajectory generated by the geometric planner in the first step. We would like to reduce this dependence and, furthermore, provide a path that is shorter and has fewer maneuvers. Because we know the form of the minimal length path between any two configurations, we simply search for collision-free paths that connect intermediate configurations on the initial feasible trajectory. We should emphasize, however, that the optimization of the trajectory is not possible in the sense that, in general, in the presence of obstacles, the infimum of path length can not be attained by a collision-free path. The minimum-length path between two configurations in the presence of obstacles is not collision-free; it will pass through the boundary of the free configuration space.

Because the notion of minimal length can only be defined locally, we present a iterative relaxation-based method that incrementally reduces the path length. Its stopping conditions are based upon a certain number of repeated failures to reduce the path length. Perhaps the best justification for the technique we propose is that, in practice, it is highly efficient at generating a subjectively pleasing trajectory in a short time.

At each iteration, two intermediate configurations along the current feasible path are generated randomly. The minimum-length feasible path connecting these two configurations, ignoring the obstacles, is then calculated. If this path is collision-free and is not the same length as the original portion of the path passing between the two configurations, it replaces that piece in the resulting feasible path. Because it is of minimal length in the absence of obstacles, we know that the new path is no longer than the original. In fact, this technique typically generates paths that are significantly shorter than the initial path. However, because we are always replacing a portion of the initial path with a minimal-length path, the length of the path is always non-increasing. Thus, the relaxation method is extremely likely to find a "locally minimal"-length path, and not a global minimum.

Remark 2: There are techniques available, such as simulated annealing, to attempt to find a globally optimal solution, for such a nonlinear constrained optimization problem. However, such techniques tend to require a great deal of time to converge, and hence we have ruled out their use in the near-term. Because the planner can operate on any collision-free trajectory, it seems more likely that gains in finding better trajectories can be achieved by adding heuristics to the initial geometric planner so that it produces an initial path that is more suited to the nonholonomic system. It is one of the strengths of this method that any heuristics added at this stage do *not* affect the completeness or exactness of the overall planner.

IV. Algorithm Analysis

A. Convergence and Completeness

Now that we have defined the basic operation of the algorithm, we need to show that, in fact, this algorithm converges and is complete; i.e., it will always find a path if one exists. The key property we need is as follows.

Given any $\epsilon > 0$, there exists $\delta > 0$ such that for any two configurations that are separated by less than δ, all the configurations along the shortest path connecting them will lie in some neighborhood of diameter ϵ of the two configurations.[8]

This result is a straightforward consequence of the topological property (see the corollary in Section II-D). It is enough to prove both convergence and completeness of the algorithm because, if one sufficiently subdivides the path provided by the geometric motion planner, then at some point a sequence of configurations will be generated that are sufficiently close that the shortest paths linking each of these configurations must necessarily lie in the free configuration space. From compactness of the initial reference path, it is always possible to provide a *finite* covering of it by neighborhoods satisfying

[6] Due to the speed of the current implementation, we have not implemented such a scheme since the preprocessing would most likely take more time than finding the entire path.

[7] We ensure that the configurations are spaced at roughly equal intervals along the path.

[8] In this case, the distance is measured in some metric considering the configuration space locally as $\mathbf{R}^3$.

this property. From this it is clear that the algorithm converges in finite time.

The completeness of the algorithm depends only on the completeness of the geometric planner used for computing the first holonomic path. Since both geometric planners described above are complete, we can conclude that the two planners are complete.

B. Complexity

Complexity of the algorithm is another question that must be answered. Here the relationship between ϵ and δ becomes important.

The algorithm splits a given holonomic path into pieces that are replaced by shortest paths. Property 2 says that in the worst case the length of the feasible path connecting two configurations is of the order of the square root of their separation measured in the Euclidean norm,[9] i.e., δ is in $O(\epsilon^{1/2})$. Therefore, if the geometric planner computes a path for which all the configurations are contained in an open ball of size ϵ lying completely in the free configuration space, then the geometric path must be cut into pieces of length at most $O(\epsilon^2)$ in order to be guaranteed that the feasible path joining them does not leave the free configuration space. Thus, in the worst case, the subdivision procedure will continue until it cuts the path into portions of this length.

Note that the worst case is necessarily reached in the case of the parking task (see Fig. 6) when we replace both front and rear "obstacle-cars" by two long vertical walls of length L; in this case the algorithm runs in $O(L/\epsilon^2)$ with ϵ being related to size of the free-space, i.e., equal to the difference between the width of the corridor and the length of the car.

What is interesting here is that the complexity of the algorithm strongly depends upon the path generated by the geometric planner in the first step and the "size" of the free space around this path. From a practical point of view, the algorithm is more efficient as the geometric path is farther from the obstacles. For instance, using a Voronoï diagram as geometric planner for the case of a circular system makes the algorithm converge very quickly (compare the running times of Step 2 in Fig. 7 and 10).

This idea has been developed by Mirtich and Canny [36] in a method closely parallel to our algorithm. Rather than compute a Voronoï diagram of the environment, the authors construct a skeleton formed by the set of configurations of maximum clearance with respect to the obstacles, where the clearance is computed in the d_{RS} metric.

Finally, the complexity of our algorithm depends also on the lower bound ρ on the turning radius. Simple geometric arguments show that, for some fixed value of ϵ, δ is in $O(1/\rho)$. This means that our algorithm runs in $O(\rho/\epsilon^2)$.

C. Comments

1) On the Number of Cusps: The optimization routine (Step 3 of the algorithm) leads to reduce the number of pieces of shortest paths provided by Step 2 of the algorithm. Moreover, shortest paths, as characterized by Reeds and Shepp, do not contain more than two cusps. Therefore, as a side effect, reducing iteratively the length of the global path reduces the number of cusps.

Nevertheless, our algorithm does not find a path with a minimum number of cusps. The problem of finding such a path is a very difficult and open one (see [5] for an approximated solution of the problem). Evaluating the computational complexity of this problem is also unsolved at this time. To our knowledge, the only existing results deal with the existence of smooth paths [14], [19] (see Appendix A for other references on planning smooth paths for a car-like robot).

Finally, the above relationship between δ and ϵ shows that the number of maneuvers for the classical car-parking task increases as the square of the decreasing free-space. As car drivers, we may verify this result everyday$\cdots$.

2) On the Shortest Paths in the Presence of Obstacles: As mentioned above, the algorithm does not find an optimal length path. Indeed, the algorithm works in components of the free configuration space (i.e., in an open set of the configuration space), while the shortest paths in the presence of obstacles usually contain some pieces in contact. At this time, the characterization of the shortest paths in such cases is unknown.

To the authors' knowledge, the only existing results deal with a characterization of the shortest smooth paths for a point or a polygon moving in the presence of polygonal obstacles [19], [20] (see Appendix A).

3) On the Motions in the Contact Configuration Space: We assumed (see Section III-A) that the environments processed by the algorithm are in general disposition. This hypothesis is realistic for practical applications of mobile robots. In this case, the number of connected components of the free space (i.e., without contact) is the same as the number of connected components of the admissible space (i.e., including contacts). It is always possible to transform a path with some pieces in contact into one that is collision- and contact-free.

In the general case, the admissible configuration space may be disconnected by the removal of the manifold consisting of configurations in contact. Thus, a solution path may necessarily contain some pieces in contact. A solution path that respects the nonholonomic constraints exists if and only if there exists a connected sub-manifold in the contact manifold whose tangent space at every point is spanned by both vector fields X_1 and $X_1 + X_2$, without requiring the direction given by $[X_1, X_1 + X_2]$. Indeed, a movement in the direction of the bracket requires an open set to move in.

Maintaining contact of the robot with the environment removes at least one degree of freedom. Therefore the dimension of the set of points reachable from a contact point by a motion in contact is at most one. This means that there exists at most one feasible path in contact passing through a contact point. For instance, it is possible to characterize the paths maintaining the contact of a robot vertex with an environment edge [20].

4) On the Complexity of Nonholonomic Motion Planning: Finally, complexity of motion planning for the car-like robot is not very well modeled at this time. The complexity analysis above gives only some partial results.

[9] Indeed, when $c = (x, y, \theta)$ is sufficiently near the origin, $|x| + |y|^{1/2} + |\theta|$ is dominated by $|y|^{1/2}$.

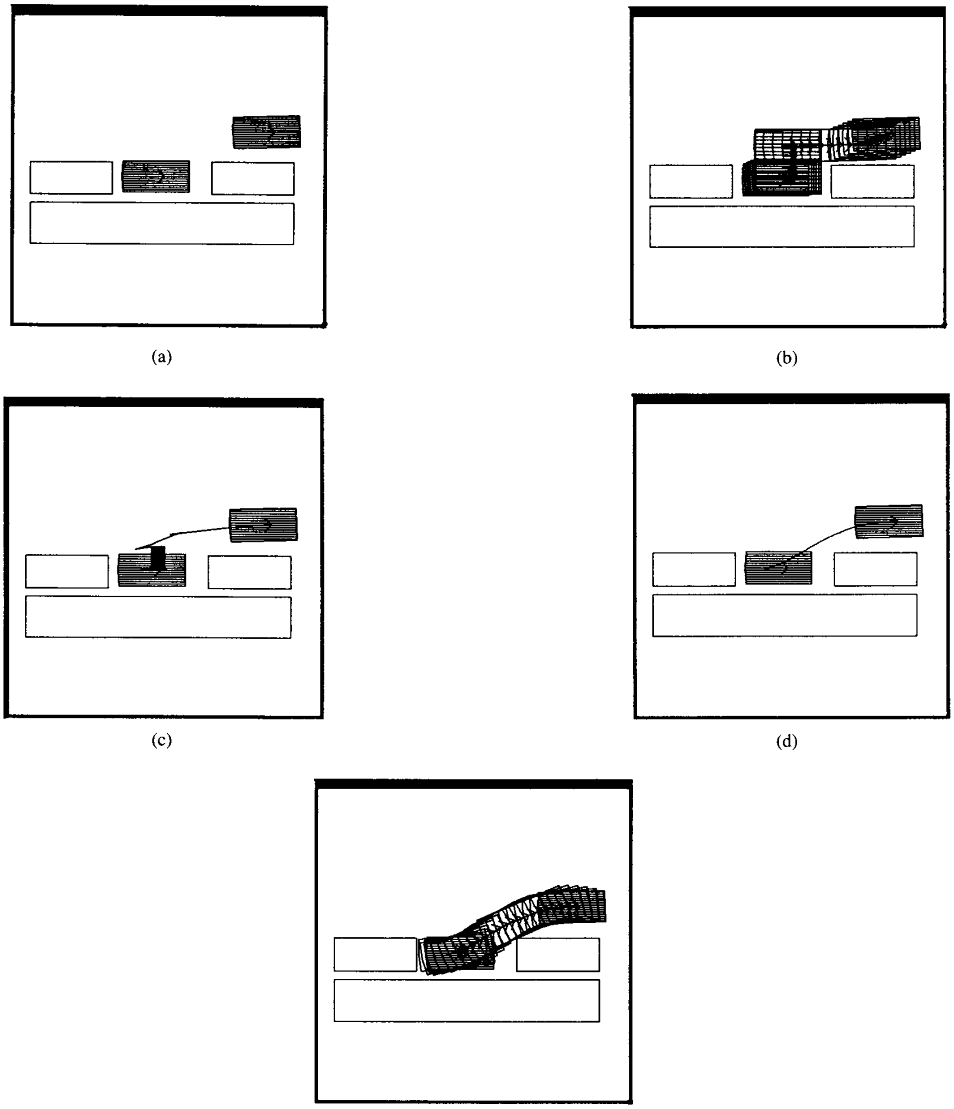
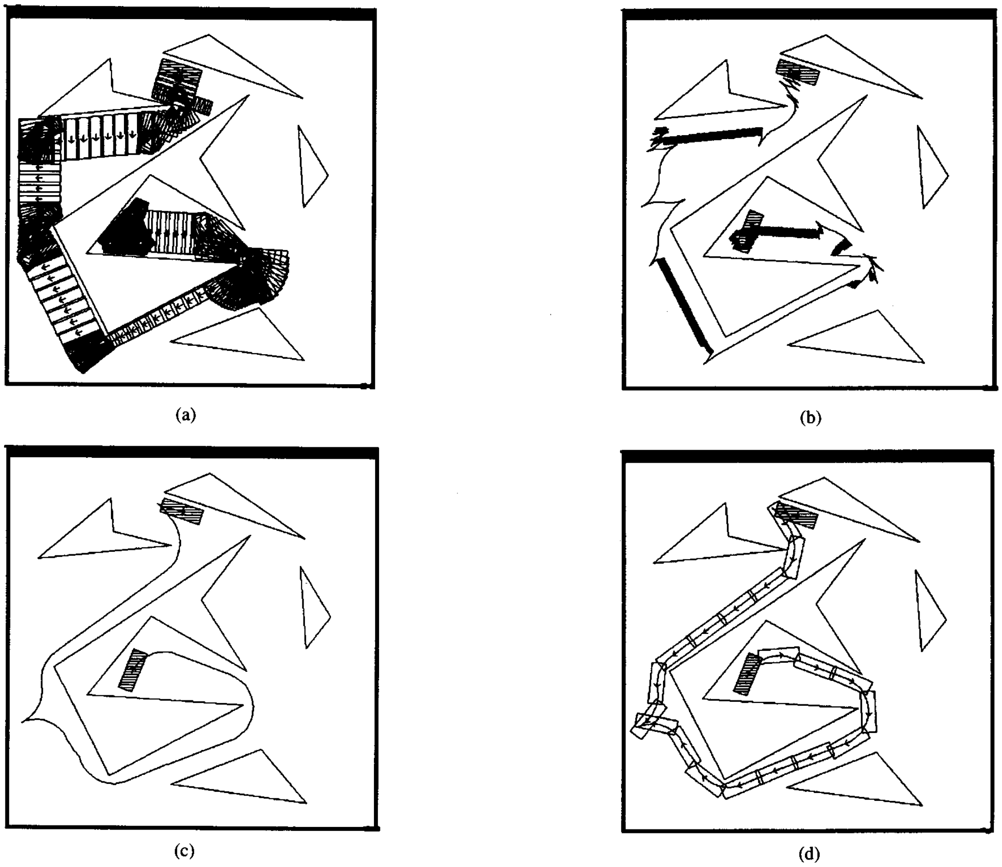


Fig. 6. The classic example. (a) Initial and final configurations. (b) Step 1: Holonomic path (0.2 s). (c) Step 2: Feasible path (0.8 s). (d) Step 3: After optimization (1.5 s). (e) Total time: 2.5 s on Sparc 2.

The complexity of the piano mover problem (i.e., the motion planning problem for holonomic systems) is very well modeled and understood. Indeed, in such cases, the existence of a path is characterized by the existence of a connected component in the admissible configuration space. The complexity of building such components depends only on the dimension of the configuration space and the complexity of the environment expressed in terms of the number of geometric primitives modeling the obstacles in the real world together with their geometric complexity. For instance, the complexity of the classical methods using a modeling of the environment by semi-algebraic sets [10], [45] are expressed in terms of the number of polynomials needed for describing the obstacles together with their degrees. Roughly speaking, the number of polynomials is related to the number of the obstacles, while the degree of the polynomials is related to the geometric complexity of their shape. Therefore, in the attractive case of polygonal obstacles the complexity of the problem is expressed from the number of obstacle vertices. Usually the complexity of deciding the existence of a connected component in the admissible configuration space dominates the complexity of the complete problem (i.e., providing a path if one exists).

In nonholonomic motion planning we have to refine the complexity model.

As noticed in Section I, the existence of a collision-free path for a controllable system is characterized by the existence of an open connected component of the admissible configuration space (up to the above subtleties dealing with motions in



(a)

(b)

(c)

(d)

Fig. 7. A more complicated example. (a) Step 1: Holonomic path (5 s). (b) Step 2: Feasible path (23 s). (c) Step 3: After optimization (10 s). (d) Total time: 38 s on Sparc 2.

contact). Therefore, the complexity of the decision part of the motion planning problem is the same for controllable systems as for holonomic ones.[10]

Complexity of the complete problem requires more developments. It has to be evaluated in terms of the following quantities:

n Geometric complexity of the obstacles.
m Geometric complexity of the robot.
ϵ Minimum "size" of the free space.
ρ Minimum turning radius.
σ "Complexity" of the output path.

n and m are the classical parameters used for evaluating the complexity of the methods solving the piano-mover problem (i.e., in our running examples n and m are the numbers of vertices of the obstacles and the robot respectively. Now, we have to precisely state the definitions of ϵ and σ.

An intrinsic definition of ϵ can be done as follows: let c be any point in the free configuration space and $B(c, \epsilon_c)$ the biggest (Riemannian) ball containing c (not necessarily centered at c), with radius ϵ_c and contained in the free space; then ϵ is defined as $\mathrm{Min}_c\{\epsilon_c\}$. Such balls are centered on a Voronoï skeleton of the free configuration space. Due to Property 2, for ϵ sufficiently small, the number of Reeds and Shepp balls of radius ϵ required to cover a Riemannian ball of radius ϵ is in $O(\epsilon^{-2})$.

At a first approach, the complexity σ of the output of any motion planner can be characterized by the number of elementary pieces of the solution path. More precisely the complexity of a path can be evaluated as the number of switches in the controls. The paths produced by our algorithm consist of a sequence of the shortest paths; since the shortest paths contain at most five pieces requiring at most four switches in the control, σ is linear in the length of the sequence. σ is in $O(\epsilon^{-2})$ and it appears that reasoning from the Voronoï diagram (as in the planner illustrated by Figure 10) is the optimal strategy in this case.

This definition of σ seems good. Nevertheless, Sussmann and Liu provide a general strategy for nonholonomic motion planning that produces solution paths with *smooth* controls (i.e., without switches) [54]. Then we have to refine the definition of σ. In [6], the complexity of a path is defined from the topological complexity of a real-value function (i.e., the number of changes in the sign of variation of the function). σ then appears as the sum of the total numbers of sign changes

[10] Notice that the problem of deciding if a system is controllable or not is not a decidable one in general [32], while upper bounds on the complexity can be done in the case of polynomial systems [7].

for all the controls associated to the path.[11] At this time, there is no result permitting to provide a relationship between σ and ϵ for Sussmann and Liu's algorithm.

In a more general perspective, providing relationships between the geometric complexity of the environment, the size of the free space and the complexity of the paths is still an open problem.

Notice here the results obtained by Fortune and Wilfong [14]: they attacked the problem of finding smooth paths for the car-like system avoiding polygonal obstacles. Their algorithm decides if a smooth path exists in $2^{O(poly(kn))}$, where n is the geometric complexity of a polygonal environment, k is the number of bits used to specify the positions of the corners, and *poly* is a polynomial. It is not clear how to link the parameter k with the parameter ϵ we have introduced.

V. Experimental Results

In this section we present experimental results from the planners that we have implemented at the LAAS. The implementation of the nonholonomic (Step 2) and optimizing (Step 3) parts of the planner consists of 2200 lines of C code running on a Sun SPARCstation 2.

We present several examples of trajectories which are generated by the planner (Figs. 6–10). We include the times required for the underlying geometric planner, but, as mentioned, any geometric planner could be used for the first step of the algorithm. For instance, examples shown in Figs. 6–9 come from the geometric planner using an exact representation of the configuration space for a polygonal robot moving amid polygonal obstacles [3]. Figure 10 shows an example from a geometric planner for a circular robot based on the Voronoï diagram [41].

Finally, another implementation of our planner has been done by Latombe [26] from a geometric planner using a bitmap representation of the environment.

VI. Conclusions

We have presented a complete and efficient motion planner for a car-like robot. The proof of the algorithm is based on a topological property involving the metric induced by the length of the shortest feasible paths in the absence of obstacles.

The principles of the algorithm can be applied to general controllable systems.

Indeed, for a controllable system, the existence of a collision-free trajectory is characterized by the existence of an open connected domain of the admissible configuration space. Therefore, the decision part of the planning problem can be solved by any geometric planner solving the piano-movers problem.

Moreover, we have seen how to put the topological property into the general framework of sub-Riemannian geometry. Such a framework is very powerful tool: indeed, the equivalence between both sub-Riemannian and Riemannian topologies is a general result.

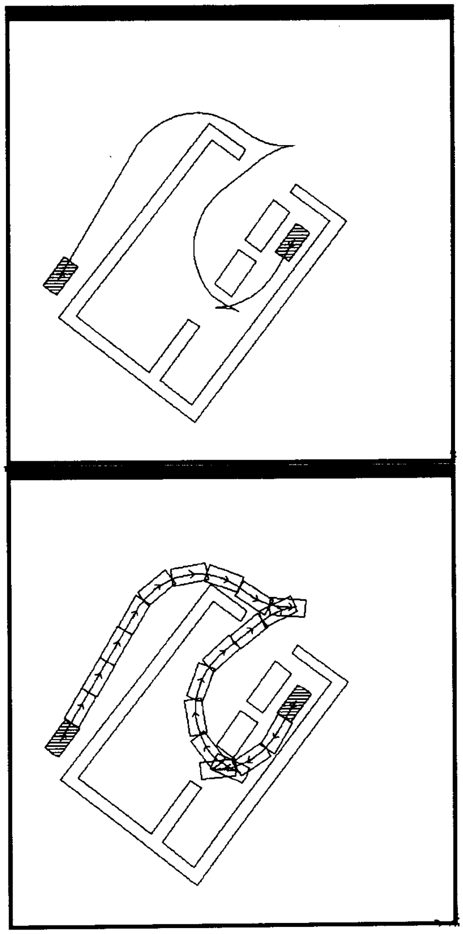


Fig. 8. A solution example with a large bound on the turning radius.

The three-step algorithm we have proposed can be applied to a general controllable system as follows:

1) Plan a path for the geometric system. If one does not exist, then no feasible path exists.
2) Perform a subdivision on the path until all endpoints can be linked by optimal feasible paths.
3) Perform an "optimization" routine to reduce the length of the path.

Such a strategy requires the knowledge of how to compute optimal feasible paths in the configuration space equipped with the singular metric induced by the nonholonomic constraints. This is a challenging requirement. It has been solved in this paper because we know the form of the shortest paths for a car-like robot in the Euclidean plane. In more general cases, time-optimal paths can be computed by using the Maximum Principle [42] in the framework of Optimal Control Theory. While this is not easy for an arbitrary nonholonomic system, recent results show that this direction of research can be fruitful. Indeed, Sussmann *et al.* [55] and Boissonnat *et al.* [8] have independently given a new proof of Reeds and Shepp's result by using the Maximum Principle, while Jacobs *et al.*

[11] This definition is more general than the previous case.

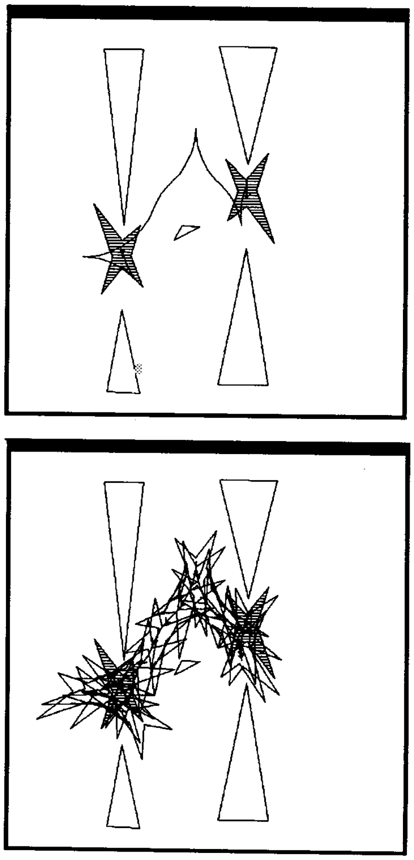


Fig. 9. A solution to a complicated situation with a concave robot.

[21] characterized the local shape of the optimal paths for mobile robots with two independent driving wheels.

Now, arguments from sub-Riemannian geometry can be applied in order to prove properties similar to Property 2 in the general cases (see Gershkovitch and Vershik's works [15], [60] that derive upper and lower bounds on the sub-Riemannian metrics). Then, we can use exactly the same techniques in order to evaluate the complexity of the motion planning algorithm from the form of the sub-Riemannian metrics. This form depends directly on the levels of the Lie brackets necessary for spanning the entire tangent space. For instance, we conjecture that the complexity of the parking task for a multi-body car system (i.e., a car with n trailers) is in $O(\epsilon^{-\frac{1}{n+1}})$ where ϵ is related to the "size" of the free-space and has exactly the same meaning as in Section IV-B. Details on these generalizations are under study (see [32]).

Appendix A
Related Work

In this appendix, we present an overview of results recently obtained in constrained motion planning. They are complementary to those discussed in the paper.

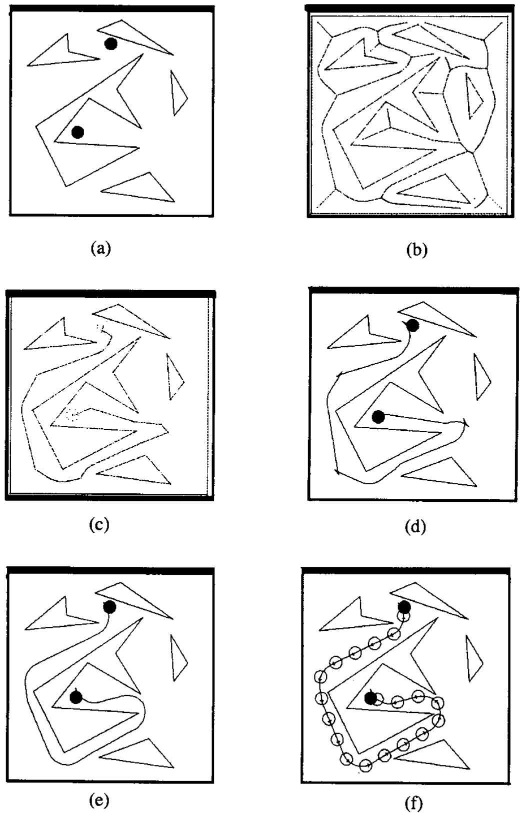


Fig. 10. The same problem as in Fig. 7 for a circular robot. (a) Initial and final configurations. (b) Collision-free Voronoï diagram (2 s). (c) Step 1: Holonomic path (0.5 s). (d) Step 2: Feasible path (2 s). (e) Step 3: After optimization (10 s). (f) Total time: 14.5 s on Sparc 2.

A. Car-Like and Trailer-Like Robots

In [27] the controllability of a car-like robot was studied. The main result is that the existence of a collision-free trajectory for such a system is characterized by the existence of a connected component in the free configuration space. The constructive proof of this result has been implemented in a planner [29]. This planner approximates the free configuration space by a decomposing it into parallelepipeds.

The planner described in [5] uses a discretization of the phase space, the adjacency relation between two cells taking into account the kinematical constraints. The collision tests are done from a bitmap representation of the 2-D environment. The search algorithm applies a best-first search strategy whose cost function is the number of maneuvers. The planner then produces paths with a minimal (for the discretized representation) number of maneuvers.

A heuristic approach was developed in [58]: the 2-D environment is decomposed into a set of corridors in which a specific technique for planning smooth paths is applied. The problem of smoothing paths is also addressed in [22] by using clothoids. Finally [62] proposes an algorithm that produces a motion with the minimum number of turns from an environment decomposed into lanes.

The case of the trailer-like robots has been attacked in [30]; the controllability of such systems is proved by applying the Lie algebra formalism briefly recalled in the introduction. A second constructive proof is also done, but does not lead to an efficient algorithm. [33] gives a proof of controllability for a mobile robot with any number of trailers.

Barraquand and Latombe's planner cited above is also applicable successfully to a trailer-like robot. Their planner is thus the first to produce paths for such a system.

B. Smooth Paths

The problem of planning smooth, maneuver-free, paths for a mobile robot appears to be more difficult than the case in which maneuvers are allowed. In fact, there is no comparable controllability result. It is indeed possible that a path exists for a holonomic system, and yet no smooth path exists. The specific problem was first addressed by Dubins [12], who gives the form of the shortest bounded curvature path in the absence of obstacles. The problem of obstacle avoidance appears more recently. In [28], the environments consist of closed curves that are not necessarily polygons. Unfortunately, the method presented there is not guaranteed to find a path. In Fortune and Wilfong [14], a decision algorithm is given to decide if a path exists under given conditions. The algorithm is exact, but does not generate the path in question. This algorithm runs in exponential time and space. The algorithm in [19], is a provably good approximation algorithm to solve the problem. The algorithm depends on simplifying the set of all smooth trajectories to a sufficient set of some canonical trajectories.

C. Shortest Paths

As mentioned above, the study of shortest paths for constrained systems began with the seminal work of Dubins [12], who found the form of the minimal-length smooth paths with bounded average curvature. He established his result in the case that there are no obstacles. The algorithm presented in [19] is a provably good approximation algorithm that depends upon an extension of the results of Dubins to the case in which there are obstacles in the environment. This extension has been presented in a rigorous manner in [20]. When there are obstacles, the minimal-length smooth paths consist of paths with the form given by Dubins alternating with possibly single-point portions of the obstacle boundaries. Finally, the same proof holds for a robot with a polygonal body moving in a polygonal environment. However, in this case the portions of the obstacle boundaries are replaced by logarithmic curves that we call generalized tractrices. These curves are those that maintain contact between an obstacle edge and a robot vertex while satisfying the nonholonomic kinematic constraint [58].

D. Other Issues in Nonholonomic Motion Planning

In addition to our obstacle avoidance context, nonholonomic motion planning appears also as an important topic in the study of holonomic systems exhibiting conservation laws. For instance, such a system is a satellite whose rotors are used in order to control its attitude. In general, equations given by conservation laws are not integrable and thus give rise to nonholonomic constraints. For instance, [40] shows how the orientation of a space vehicle with a six-degree-of-freedom manipulator can be controlled. Optimal movements of nonholonomically constrained multibody systems floating in zero gravity are considered in [23]. [13] studies other representative systems (rolling disk, satellite with rotors, falling cat$\cdots$) and uses variational principles to characterize optimal solutions to the associated nonholonomic motion planning problems.

APPENDIX B
PROOFS OF PROPERTY I

This appendix presents two proofs of Property 1 concerning the controllability of the car-like system. The first one can be read without reference to differential geometric control theory. Following this, we introduce basic concepts from differential geometry. Their power is seen in a second version of the proof, which illustrates the general proof of the controllability theorem given by the Lie algebra rank condition. This is the goal of this appendix.

A. Direct Proof

The notion of controllability we use is defined from the local controllability notion in control theory [53]: a system is *locally controllable* from some point c, if there is a neighborhood of c all of whose points are reachable from c by an admissible path. Here, we say that the system is *controllable* if it is locally controllable at every point.

Property 1: The car-like system is controllable.

Proof 1: It suffices to prove that the system is locally controllable from the origin. Let $c = (x, y, \theta)$ be a point near the origin. The idea of the proof is to build a path consisting of three pieces γ_1, γ_2, and γ_3 attaining from c, in succession, the points $c_1 = (x_1, y_1, 0), c_2 = (x_2, 0, 0)$ and the origin. Clearly, this path can be followed by the car-like system in the reverse direction to go from the origin to c.

Let γ_1 be the arc of the circle[12] tangent at c, with length $|\theta|$ following in the direction of $-\text{sign}(\theta)$. We assume that $\theta \geq 0$. γ_1 attains the point $c_1 = (x - \sin\theta, y - (1 - \cos\theta), 0)$. We assume that $y - (1 - \cos\theta) > 0$. Other cases would be processed in the same way (see Fig. 11).

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

$$|y - (1 - \cos\theta)| = \tau_c \sin\tau_c \tag{7}$$

Such τ_c always exists (and is unique) for any c sufficiently near the origin. The coordinates of point c_2 attained by $\gamma(\tau_c)$ are: $(x - \sin\theta + \tau_c(1 - \cos\tau_c), 0, 0)$.

Finally, let γ_3 be the straight line motion from c_2 to the origin.

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

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

The length of the path built in the proof above is

$$l = |\theta| + 4\tau_c + |x - \sin\theta + \tau_c(1 - \cos\tau_c)| \qquad (8)$$

The proof of Property 2 below will give an upper bound for this length. This means that, for any time T there exists a neighborhood of the origin such that all the points in this neighborhood can be reached from the origin in time bounded by T. In this case, the system is said to be *small-time locally controllable* [53] from the origin.

B. From Vector Fields to Paths

The concepts from differential geometry that we want to introduce here are thoroughly studied in [49]–[51]. They will be used in order to give another method for proving Property 2. The difference between the two proofs is that the second is more general, illustrating the power of the differential geometric control theory approach.

Choose a point p on a manifold and a vector field X defined around this point. There is exactly one path $\gamma(t)$ starting at this point and following X. That is, it satisfies $\gamma(0) = p$ and $\dot\gamma(t) = X_{\gamma(t)}$. One defines the exponential of X (denoted by e^X) to be the point $\gamma(1)$. Now, one can consider $e^{tX} = \gamma(t)$, where this definition doesn't describe a peculiar point of a path, but, more accurately, links every point of the path to a specific vector field. Let us consider two vector fields X and Y. Following $aX + bY$ for the unit time simply means taking e^{aX+bY}. Following X for a time a amounts to following aX for the unit time, that is taking e^{aX}, and following Y for a time b is the same as taking e^{bY}. This is a slightly different point of view: instead of considering the exponential to define a specific point of a path with regard to an origin point p, we understand it as describing a motion from a point to another on a given path. Thus, starting from the origin, o, following aX for a given time, then bY leaves us at the point $e^{bY} \cdot e^{aX} \cdot o$. Therefore the exponential of a vector field X appears as an operation on the manifold, meaning "slide from the given point along the vector field X for unit time."

In that setting, everything works nearly as smoothly as in the Euclidean case, at least locally. The main difference is that, whenever $[X, Y] \neq 0$, following directly $aX+bY$ or following first aX then bY are no longer equivalent. Intuitively, the bracket $[X, Y]$ measures the variation of Y along the paths of X; in other words, the field Y we follow in $aX + bY$ is not the same as the field Y we follow after having followed only aX (indeed Y is not evaluated at the same points in both cases). The main result is the following:

Assume that $X_1, \cdots, X_n$ are vector fields defined in a neighborhood $\mathcal{N}$ of a point p such that at each point of $\mathcal{N}$, $\{X_1, \cdots, X_n\}$ constitutes a basis of the tangent space. Then there is a smaller neighborhood of p on which the maps $(a_1, \cdots, a_n) \mapsto e^{a_1X_1+\cdots+a_nX_n} \cdot p$ and $(a_1, \cdots, a_n) \mapsto e^{a_nX_n} \cdots e^{a_1X_1} \cdot p$ are two coordinate systems, called the first and the second normal coordinate system associated to $\{X_1, \cdots, X_n\}$.

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

$$e^{tX} \cdot e^{tY} = e^{tX+tY-(1/2)t^2[X,Y]+t^2\epsilon(t)}$$

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

Actually, the whole formula as proved in [59] gives an explicit form for the ϵ function. More precisely, ϵ yields a formal series whose coefficients c_k of t^k are combinations of brackets of degree k,[13] i.e.,

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

Roughly speaking, the Campbell-Baker-Hausdorff-Dynkin formula tells us how a 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 a method for *explicitly computing paths* in a neighborhood of a point. We now illustrate this point with another proof of Property 1.

C. Proof Using Campbell-Baker-Hausdorff-Dynkin Formula

Property 1: The car-like system is controllable.

Proof 2: We use the notations introduced in Section II. Let us consider the two admissible controls:

$$\begin{pmatrix}1\\0\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}1\\1\end{pmatrix}.$$

The associated vector fields are X_1 and $X_1 + X_2$. The coordinates of $X_1, X_1 + X_2$ and $[X_1, X_1 + X_2]$ are, respectively,

$$X_1 = \begin{pmatrix}\cos\theta\\ \sin\theta\\ 0\end{pmatrix}, \quad X_1 + X_2 = \begin{pmatrix}\cos\theta\\ \sin\theta\\ 1\end{pmatrix},$$

$$\text{and} \quad [X_1, X_1 + X_2] = \begin{pmatrix}-\sin\theta\\ \cos\theta\\ 0\end{pmatrix}.$$

We check easily that $\{X_1, [X_1, X_1 + X_2], X_1 + X_2, \}$ spans the tangent space at every point. Let us prove that the system is locally controllable from the origin. The same proof would hold for every point.

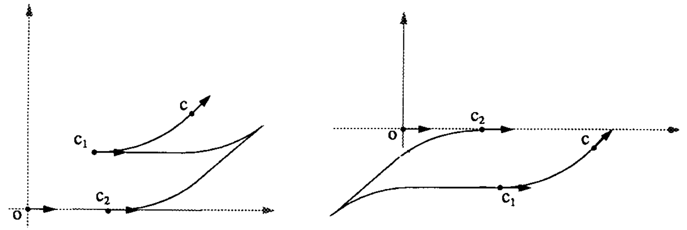
Consider a point c in a neighborhood of the origin. Let (t_1, t_2, t_3) be its coordinates in the coordinate system induced by the basis $\{X_1, [X_1, X_1 + X_2], X_1 + X_2, \}$, which spans the tangent space at the origin. c is reachable from the origin by the following flow:

$$\phi(t_1, t_2, t_3) = e^{t_1X_1}e^{t_2[X_1,X_1+X_2]}e^{t_3(X_1+X_2)}. \qquad (9)$$

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

$$e^{(t[X_1,X_1+X_2]+O(t^{3/2}))} = e^{t^{1/2}X_1}e^{t^{1/2}(X_1+X_2)} e^{-t^{1/2}X_1}e^{-t^{1/2}(X_1+X_2)}. \qquad (10)$$

[13] As an example, the degree of $[[X, Y], [X, [Y, [X, Y]]]]$ is 6.



Case : θ > 0 and y - (1 - cos θ) > 0
(a)

Case : θ > 0 and y - (1 - cos θ) < 0
(b)

Fig. 11. Two ways to reach the origin according to sign(θ) and sign($y - (1 - \cos\theta)$).

This shows that any configuration we can attain by following the flow given in (9) can be approximated by the following flow, which obeys the constraints:

$$\tilde{\phi}(t_1, t_2, t_3) = e^{t_1 X_1} e^{t_2^{1/2} X_1} e^{t_2^{1/2}(X_1+X_2)} e^{-t_2^{1/2} X_1} e^{-t_2^{1/2}(X_1+X_2)} e^{t_3(X_1+X_2)}. \tag{11}$$

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

In fact, for the car-like robot, the differential equations are sufficiently simple that they can be integrated in closed form. This is far from being true in general. More precisely, the flow $\tilde{\phi}$ corresponds *exactly* to the path provided in Proof 1 and followed by starting from the origin. It attains a point of coordinates:

$$\begin{aligned} x &= t_1 + \operatorname{sign}(t_2)\sqrt{|t_2|}(1 - \cos(\sqrt{|t_2|})) + \sin(t_3) \\ y &= -\operatorname{sign}(t_2)\sqrt{|t_2|}\sin(\sqrt{|t_2|}) + 1 - \cos(t_3) \\ \theta &= t_3. \end{aligned} \tag{12}$$

By expressing this point in the basis $\{X_1, [X_1, X_1+X_2], X_1+X_2,\}$ we obtain the asymptotic estimates $(t_1+O(t_2^{3/2}), t_3, t_2+O(t_2^{\frac{3}{2}}))$.

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

Appendix C
Proof of Property 2

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

Property 2: For any point $c = (x, y, \theta)$ sufficiently near the origin $o = (0, 0, 0)$,

$$\tfrac{1}{3}(|x| + |y|^{1/2} + |\theta|) \leq d_{RS}(c, o) \leq 12(|x| + |y|^{1/2} + |\theta|).$$

Proof: 1) We first prove the left part of the inequalities. Let us consider any admissible path γ parametrized by arc length in $\mathbf{R}^2$ such that $\gamma(0) = o$. Let us denote by $c(t)$ the point with coordinates $(x(t), y(t), \theta(t))$ attained by an admissible path of length t. Since the path is admissible, then, for each $\tau \in [0, t], \dot{\gamma}(\tau)$ is a linear combination of the vector fields X_1 and X_1+X_2, i.e.: $\dot{\gamma}(\tau) = \lambda(\tau)X_1(\tau)+\mu(\tau)(X_1(\tau)+X_2(\tau))$, with $|\lambda(\tau)| \leq 1$ and $|\mu(\tau)| \leq 1$. Then:

$$\begin{aligned} |y(t)| &\leq \int_0^t |\dot{y}(\tau)|\, d\tau \\ &= \int_0^t |(\lambda(\tau) + \mu(\tau)) \cdot \sin\theta(\tau)|\, d\tau \\ &\leq 2\int_0^t |\sin\theta(\tau)|\, d\tau. \end{aligned}$$

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

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

Finally:

$$|y(t)| \leq 2\int_0^t \tau\, d\tau = t^2. \tag{13}$$

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

$$|y|^{1/2} \leq d_{RS}(c, o). \tag{14}$$

Now, the following inequalities hold:

$$|x| \leq d_{RS}(c, o) \quad \text{and} \quad |\theta| \leq d_{RS}(c, o). \tag{15}$$

Then, by adding inequalities (14) and (15), we obtain the following lower bound:

$$\tfrac{1}{3}(|x| + |y|^{1/2} + |\theta|) \leq d_{RS}(c, o).$$

2) The second part of the proof deals with the right side of the inequalities. We consider the path built in Proof 1 of

Property 1 starting from a point $c = (x, y, \theta)$ near the origin to the origin. The length l of this path is (see (8) above):

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

with τ_c satisfying (7). A first upper bound of l is clearly done by:

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

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

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

Now, let us consider (7): $|y - (1 - \cos\theta)| = \tau_c \sin\tau_c$. For c sufficiently near the origin, τ_c is sufficiently small for verifying $\sin\tau_c \geq (1/4)\tau_c$. Then we may write $\tau_c \leq 2|y - (1 - \cos\theta)|^{1/2}$. Moreover, $|1 - \cos\theta| \leq \theta^2$ for θ sufficiently small. Finally,

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

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

ACKNOWLEDGMENT

The authors would like to thank A. Bellaïche for useful discussions on sub-Riemannian geometry, P. Souères for checking the proofs in the appendixes, and the anonymous referees for their valuable comments. Finally, the authors thank G. Giralt for discussions that initiated this work.

REFERENCES

[1] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications.* Reading, MA: Addison-Wesley Publishing Company, Inc., 1983.

[2] B. d'Andréa-Novel, G. Bastin, and G. Campion, "Dynamical feedback linearization of nonholonomic wheeled mobile robots," in *IEEE Conf. Robotics Automat.*, Nice, France, 1992, pp. 2527–2532.

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

[4] F. Avnaim, "Placement et déplacement de formes rigides ou articulées," Ph.D. thesis, Univ. Franche-Comté, 1989.

[5] J. Barraquand and J.-C. Latombe, "On non-holonomic mobile robots and optimal maneuvering," *Revue d'Intelligence Artificielle*, vol. 3, no. 2, pp. 77–103, 1989.

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

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

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

[9] C. Canudas de Wit and R. Roskam, "Path following of a 2-DOF wheeled mobile robot under path and input torque constraints," in *IEEE Int. Conf. Robotics Automat.*, Sacramento, April 1991, pp. 1142–1147.

[10] J. F. Canny, *The complexity of robot motion planning.* Cambridge, MA: MIT Press, 1998.

[11] R. Chatila, "Mobile robot navigation: Space modeling and decisional processes," in *Robotics Research: The Third International Symposium*, O. Faugeras and G. Giralt, Eds. Cambridge, MA: MIT Press, pp. 373–378, 1986.

[12] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *Am. J. Math.*, vol. 79, pp. 497–516, 1957.

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

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

[15] V. Y. Gershkovich, "Two-sided estimates of metrics generated by absolutely nonholonomic distributions on Riemannian manifolds," *Soviet Math. Dokl.*, vol. 30, no. 2, pp. 506–510, 1984.

[16] G. Giralt, R. Chatila, and M. Vaisset, "An integrated navigation and motion control system for autonomous multisensory mobile robots," in *Robotics Research: The First International Symposium*, M. Brady and R. P. Paul, Eds. Cambridge, MA: MIT Press, 1984, pp. 191–214.

[17] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Trans. Automat. Contr.*, vol. 22, no. 5, pp. 728–740, 1977.

[18] G. Jacob, "Lyndon discretization and exact motion planning," in *European Contr. Conf.*, Grenoble, France, 1991, pp. 1507–1512.

[19] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *IEEE Int. Conf. Robotics and Automat.*, 1989, pp. 2–7.

[20] Paul Jacobs, "Minimal length curvature constrained paths in the presence of obstacles," LAAS/CNRS Report 90042, Toulouse, February 1990.

[21] P. Jacobs, A. Rege, and J. P. Laumond, "Non-holonomic motion planning for Hilare-like robots," in *Int. Symp. Intelligent Robotics*, Bangalore, India, January 1991, pp. 338–347.

[22] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," in *Robotics Research*, vol. 3. Cambridge, MA: MIT Press, pp. 333–340, 1986.

[23] P. S. Krishnaprasad and R. Yang, "Geometric phases, anholonomy, and optimal movement," in *IEEE Int. Conf. Robotics Automat.*, Sacramento, CA, April 1991, pp. 2185–2189.

[24] G. Lafferriere and H. J. Sussman, "Motion planning for controllable systems without drift: A preliminary report," Tech. Rep. SYSCON-90-04, Rutgers Center for Systems and Control, June 1990.

[25] J. C. Latombe, *Robot Motion Planning.* New York: Kluwer Academic Pub., 1991.

[26] J. C. Latombe, "A fast path planner for a car-like indoor mobile robot," in *Ninth Nat. Conf. Artificial Intelligence, AAAI*, Anaheim, CA, July 1991, pp. 659–665.

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

[28] J. P. Laumond, "Finding collision-free smooth trajectories for a non-holonomic mobile robot," in *10th Int. Joint Conf. Artificial Intelligence*, Milano, Italy, 1987, pp. 1120–1123.

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

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

[31] J. P. Laumond, M Taïx, and P. Jacobs, "A motion planner for car-like mobile robots based on a mixed global/local approach," in *IEEE Int. Workshop Intelligent Robots Systems*, July 1990, pp. 765–773.

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

[33] J. P. Laumond, "Controllability of a multibody mobile robot," *IEEE Trans. Robotics Automat.*, vol. 9, no. 6, pp. 755–763, 1993.

[34] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Trans. Robotics Automat.*, vol. 6, pp. 62–72, 1990.

[35] C. Lobry, "Controlabilité des systèmes non linéaires," *SIAM J. of Control*, vol. 8, pp. 573–605, 1979.

[36] B. Mirtich and J. Canny, "Using skeletons for nonholonomic path planning among obstacles," in *IEEE Int. Conf. Robotics Automat.*, Nice, May 1992.

[37] R. M. Murray and S. Sastry, "Grasping and manipulation using multifingered robot hands," in *Mathematical Question in Robotics*, Lecture Notes. New York: American Mathematical Society, 1990.

[38] R. M. Murray and S. Sastry, "Steering nonholonomic systems using sinusoids," in *Proc. of the CDC*, pp. 2097–2101, 1990.

[39] R. M. Murray, "Robotic control and nonholonomic motion planning," Ph.D. thesis, Memorandum No. UCB/ERL M90/117, University of California, Berkeley, December 1990.

[40] Y. Nakamura and R. Mukherjee, "Nonholonomic path planning of space robots via bi-directional approach," in *IEEE Int. Conf. Robotics Automat.,* 1990, pp. 1764–1769.

[41] C. O'Dunlaing and C. K. Yap, "A retraction method for planning the motion of a disc," *J. Algorithms,* vol. 6, no. 1, pp. 104–111, 1985.

[42] L. Pontryagin *et al.*, "The mathematical theory of optimal processes," in *L. S. Pontryagin Selected Works, Vol. 4.* New York: Gordon and Breach Sc. Pub., 1986.

[43] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific J. Mathematics,* vol. 145, no. 2, pp. 367–393, 1990.

[44] C. Samson and K. Ait-Abderrahim, "Feedback control of a non holonomic wheeled cart in Cartesian space," in *IEEE Int. Conf. Robotics Automat.,* Sacramento, April 1991, pp. 1136–1141.

[45] 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 Math.*, vol. 4, pp. 298–351, 1983.

[46] M. Sharir, "Algorithmic motion planning in robotics," Tech. Rep. 392, Courant Institute, New York University, 1988.

[47] P. Souères and J.-P. Laumond, "Shortest path synthesis for a car-like robot," *European Control Conf.*, Groningen, June 1993, pp. 570–577.

[48] M. Spivak, *A Comprehensive Introduction to Differential Geometry,* volume 1. Berkeley, CA: Publish or Perish, 1979.

[49] S. Sternberg, *Lectures on Differential Geometry.* New York: Chelsea Pub., 1983.

[50] R. S. Strichartz, "Sub-Riemannian geometry," *J. Differential Geometry,* vol. 24, pp. 221–263, 1986.

[51] R. S. Strichartz, "The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations," *J. Functional Analysis,* vol. 72, pp. 320–345, 1987.

[52] H. J. Sussmann and V. Jurdjevic, "Controllability of nonlinear systems," *J. Differential Equations,* vol. 12, pp. 95–116, 1972.

[53] H. Sussmann, "Lie brackets, real analyticity, and geometric control," in *Differential Geometric Control Theory,* R. Brockett, R. Millman, and H. Sussmann, Eds., in *Progress in Mathematics,* vol. 27. Michigan Technological University, Birkhauser, June 28–July 2, 1982, pp. 1–116.

[54] 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, February 1991.

[55] H. J. Sussmann and W. Tang, "Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control," Report SYCON-91-10, Rutgers University, 1991.

[56] M. Taïx, "Planification de mouvements pour robots mobiles nonholonomes," Ph.D. Thesis 824, Paul Sabatier Univ., Toulouse, January 1991.

[57] D. Tilbury, J.-P. Laumond, R. Murray, S. Sastry, and G. Walsh, "Steering car-like systems with trailers using sinusoids," in *IEEE Conf. Robotics Automation,* pp. 1993–1998, Nice, France, 1992.

[58] P. Tournassoud, "Motion planning for a mobile robot with a kinematic constraint," in *Geometry and Robotics,* J. D. Boissonnat and J. P. Laumond, Eds. in *Lecture Notes in Computer Science,* vol. 391. New York: Springer Verlag, 1989, pp. 150–171.

[59] V. S. Varadarajan, *Lie Groups, Lie Algebra, and their Representations.* New York: Springer-Verlag, 1984.

[60] A. M. Vershik and V. Ya. Gershkovich, "Nonholonomic problems and the theory of distributions," *Acta Applicandae Mathematicae,* vol. 12, pp. 181–209, 1988.

[61] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J.-P. Laumond, "Stabilization of trajectories for systems with nonholonomic constraints," in *IEEE Int. Conf. Robotics Automation,* Nice, France, 1992, pp. 1999–2004.

[62] G. T. Wilfong, "Motion planning for an autonomous vehicle," in *IEEE Int. Conf. Robotics Automation,* 1988, pp. 529–533.

Jean-Paul Laumond is Directeur de Recherche at LAAS-CNRS in Toulouse. He received the M.S. degree in Mathematics, the Ph.D. in Robotics and the Habilitation from the University Paul Sabatier at Toulouse in 1976, 1984, and 1989, respectively. From 1976 to 1983 he was teacher in Mathematics. He joined CNRS in 1985 as Chargé de Recherche at LAAS-CNRS. In 1990 he was an invited senior scientist at Stanford University. He is currently the coordinator of the European Esprit 3 Basic Research project 6546 PROMotion (Planning RObot Motion). His research interests include mainly robotics and algorithmic motion planning.

Paul E. Jacobs (S'83–M'89) was born in Cambridge, Massachusetts on October 30, 1962. He received the B.S. degree, with highest honors, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley in 1984, 1986, and 1989, respectively. He spent a postdoctoral year at the Laboratoire d'Analytique et d'Analyse des Systèmes (LAAS-CNRS) in Toulouse, France, conducting research in motion planning under nonholonomic constraints. In September 1990 he joined QUALCOMM Inc., where he holds the title of staff engineer and manager. He is currently involved in speech compression research and is technical lead of the team that developed the speech codec for the CDMA digital cellular standard.

He is a member of Phi Beta Kappa, Tau Beta Pi, and Eta Kappa Nu.

Michel Taïx graduated from the Institut National des Sciences Appliqueés (1985). He received the Ph.D. degree in robotics from the University Paul Sabatier at Toulouse in 1991. He is currently an Assistant Professor of Automation Department at the University Paul Sabatier. He is completing his research at the Laboratoire d'Automatique et d'Analyse des Systèmes of CNRS in Toulouse. His research interests are motion planning, computational geometry, mobile robot, nonholonomic systems.

Richard M. Murray (S'83–M'90) received the B.S. degree in electrical engineering from the California Institute of Technology in 1981 and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1988 and 1991, respectively. He held a postdoctoral appointment at University of California, Berkeley, in 1991 and is currently an Assistant Professor of Mechanical Engineering at the California Institute of Technology, Pasadena. His research interests include nonlinear control of mechanical systems and robotics.