Algorithmica (1993) 10: 121–155

Algorithmica

Nonholonomic Multibody Mobile Robots: Controllability and Motion Planning in the Presence of Obstacles[1]

Jérôme Barraquand[2,3] and Jean-Claude Latombe[3]

Abstract. We consider mobile robots made of a single body (car-like robots) or several bodies (tractors towing several trailers sequentially hooked). These robots are known to be nonholonomic, i.e., they are subject to nonintegrable equality kinematic constraints involving the velocity. In other words, the number of controls (dimension of the admissible velocity space), is smaller than the dimension of the configuration space. In addition, the range of possible controls is usually further constrained by inequality constraints due to mechanical stops in the steering mechanism of the tractor. We first analyze the controllability of such nonholonomic multibody robots. We show that the well-known Controllability Rank Condition Theorem is applicable to these robots even when there are inequality constraints on the velocity, in addition to the equality constraints. This allows us to subsume and generalize several controllability results recently published in the Robotics literature concerning nonholonomic mobile robots, and to infer several new important results. We then describe an implemented planner inspired by these results. We give experimental results obtained with this planner that illustrate the theoretical results previously developed.

Key Words. Path planning, Robotics, Mobile robots, Controllability, Nonholonomy, Optimal maneuvering, Collision avoidance.

1. Introduction. We consider mobile robots made of a sequence of one or several hinged bodies rolling on a flat ground among obstacles (e.g., a luggage carrier in an airport facility). The first body of the sequence is called the *tractor* and the other bodies (if any) are called the *trailers*. We call such robot a *multibody mobile robot*. We model it as a sequence of hinged rectangles (or any other two-dimensional geometric object) moving in the plane, as shown in Figure 1. We first analyze the controllability of multibody mobile robots, and then we describe an operational path planner whose design derives from this analysis. We describe experiments conducted with this planner.

---

[1] This research was partially funded by DARPA contract DAAA21-89-C0002 (Army), CIFE (Center for Integrated Facility Engineering), and Digital Equipment Corporation.
[2] Jérôme Barraquand is now with the Paris Research Laboratory of Digital Equipment Corporation (DEC), 85 Avenue Victor Hugo, 92563 Rueil–Malmaison Cedex, France.
[3] Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305, USA.

Received April, 1991; revised December, 1991. Communicated by Bruce Randall Donald.

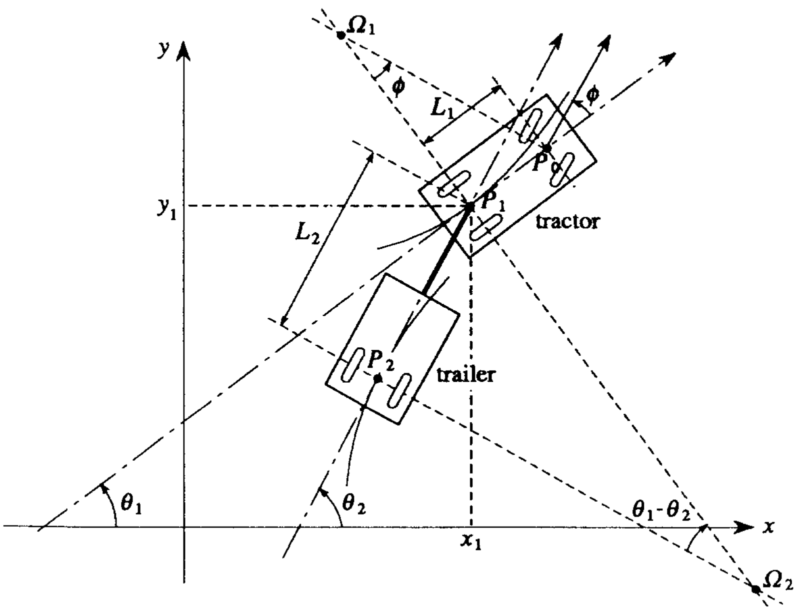

Fig. 1. Two-body mobile robot.

The motions of a multibody mobile robot are constrained by two kinds of kinematic constraints involving the velocity:

Equality constraints, which are caused by the contacts between the wheels and the ground (we assume these contacts to be pure rolling contacts between rigid objects). They express the fact that the relative velocity of two points in contact is zero. It can be shown that these constraints are nonintegrable. They make the dimension of the space of achievable (admissible) velocities smaller than the dimension of the robot's configuration space. They are called *nonholonomic equality constraints.*

Inequality constraints, which are caused by mechanical stops in the steering mechanism of the tractor. Unlike equality constraints, they do not reduce the dimension of the set of achievable velocities. Nevertheless, they locally restrict the achievable directions for the velocity to a subset of the tangent space of the configuration space at the current configuration of the robot.

For example, consider a regular automobile car (one-body robot). In the absence of obstacles, it can attain any position in the plane with any orientation. Hence, the configuration space is three dimensional. However, assuming no slipping of the wheels on the ground, the velocity of the midpoint between the two rear wheels of the car is always tangent to the car orientation. The space of achievable velocities at any configuration is thus two dimensional. This corresponds to an equality constraint on the velocity. If, in addition, the steering angle of the front wheels is limited by mechanical stops, the space of achievable velocities at any configuration is further restricted to a two-dimensional cone around the neutral position. This corresponds to an inequality constraint on the velocity.

Although multibody mobile robots are the primary focus of this paper, there exist other important examples of nonholonomic robots addressed in the Robotics literature, including levitating, undersea, and flying robots actuated by thrusters [2], [39], and dextrous hands when the spherical tips of their fingers perform rolling motions in contact with an object [12], [10], [35]. Some of our results are quite general and potentially apply to many different nonholonomic robots.

Path planning consists of constructing a path joining two input configurations in the free subset of the robot's configuration space, i.e., the set of configurations where the robot has no intersection or contact with the obstacles. Any free path, however, is not feasible. Kinematic constraints also require that the tangent to the path at any configuration be within the subspace of achievable velocities selected by the constraints. Hence, a free path for a nonholonomic robot typically includes "reversals," i.e., backing-up configurations (cusps) where the robot stops and changes the sign of the velocity (think, e.g., of the parallel parking of a car along a sidewalk).

Finding a feasible free path for a nonholonomic robot is much more difficult than finding a free path for a holonomic robot having the same geometry and the same dimension of configuration space [29], [32], [6], [7], [34]. Path planning for a nonholonomic robot directly relates to the controllability issue: Do the equality and inequality kinematic constraints restrict the set of configurations achievable by the robot? A robot is said to be *controllable* iff, for any distribution of the obstacles in the workspace, there exists a free path between two configurations, then there also exists a feasible free path between the same two configurations. Showing that a robot is controllable is a first essential step toward building a planner for that robot. It, nevertheless, remains to conceive a method for actually constructing feasible free paths. Furthermore, in order to be satisfactory, this method should produce paths that include reasonable number of reversals. This property is particularly important, since reversals cannot be smoothed out at execution time and, hence, require the robot to stop.

The main results of this paper are presented in Sections 4, 5, and 6.

In Section 4, we generalize the mathematical analysis of nonholonomic constraints previously presented in [6] and [7]. Using standard results in nonlinear control theory (namely, the Controllability Rank Condition Theorem for nonlinear systems), we state a general result applicable to robots subject to equality and/or inequality kinematic constraints involving the velocity. These constraints may be linear or nonlinear in the velocity parameters. From there, we infer new results on the controllability of nonholonomic robots.

In Section 5, we apply these results to multibody mobile robots. We show that a multibody mobile robot that can go forward and backward is controllable whenever there are at least two different admissible positions of the steering wheel of the tractor. This includes the classical case where the tractor has a lower-bounded turning radius and the more exotic case where the tractor can only turn to the left (or to the right). Hence, if there exists a free path for a multibody mobile robot whose steering angle is constrained to be within some interval, then there exists another path that uses only the extremal values of the steering angle.

In Section 6, we describe an implemented planner that searches a finite space obtained by discretizing the controls of the robot according to the above theoretical result. For any given problem that admits a solution path, it is guaranteed to find a path with minimal number of reversals, provided that the discretization parameters have been set fine enough. The planner is approximate in the sense that it produces a path whose final configuration is only contained in a neighborhood of the goal configuration. This neighborhood can be set as small as one wishes. The time/space complexity of the planner is exponential in the number of bodies. For one-body and two-body mobile robots, the planner is relatively fast and has solved a variety of nontrivial problems in reasonable time.

Before presenting these results, we relate our work to other research on nonholonomic robots (Section 2), and we provide some background in nonlinear control systems (Section 3).

Possible applications of the results presented in this paper include navigation of autonomous robots, automated parking of personal cars and trucks, autonomous navigation of luggage carrier in airport facilities, automatic planning of the movements of machines in a construction site, and computer-aided design of access ports for trucks in industrial and commercial facilities.

2. Relation to Other Work. Research on collision-free path planning has been very active during the past 10 years (e.g., see [26]). Today, the mathematical and computational structures of this problem for holonomic robots are reasonably well understood. Practical planners have also been implemented in more or less specific cases [9], [16], [28], [13], [14], [38], [4]. In contrast, the interest in path planning with nonholonomic constraints is more recent.

The problem was first introduced in the Robotics literature by Laumond [29]. Laumond proved that the single-body mobile robot is controllable. His proof is based on the definition and combination of two basic maneuvers (standard paths including several reversals). One maneuver allows the robot to move sidewise, while the other makes it rotate with a zero turning radius. Each maneuver describes a path in configuration space that can be enclosed in an arbitrarily small open set. This proof was later used to design an actual two-phase planner. In the first phase, the planner generates a free path by ignoring the nonholonomic constraints. In the second phase, it transforms this path into a topologically equivalent free path that satisfies the nonholonomic constraints by introducing maneuvers of the above types [31]. However, the number of maneuvers generated by this planner may be very large, even when there exist feasible free paths with no or few reversals. Using the same idea and a result by Reeds and Shepp [42] establishing the shape of the shortest feasible paths for a car-like robot in the absence of obstacles, a three-phase planner for car-like robots has recently been developed [44], [23] and reused in [27]. In the first phase, the planner generates a free path in the configuration space. In the second phase, it recursively decomposes this path into subpaths, until all the subpaths can be replaced by free shortest feasible paths. The outcome of the second phase is thus a free feasible path. In the third phase, the planner attempts to optimize the path by replacing feasible subpaths (randomly selected) of the current path by free shortest feasible paths. An interesting property of this algorithm is that, when the robot and the obstacles are described as polygons, it is both complete, i.e., it generates a path whenever one exists and returns failure otherwise, and exact, i.e., it generates a path that exactly connects the initial and goal configurations. But the number of reversals in the output paths may be far from optimal since the transformations carried out in the second and third phases are essentially local operations.

Another planning approach has been specifically developed and implemented for one-body mobile robots. It consists of planning the motion of the robot in a network of "corridors" [45] or "lanes" [47] that are extracted from the workspace in a first phase of processing. Local planning techniques are then used to generate turns for transferring the robot from a corridor (or lane) to another at a "crossroad" of the network. The difficulty of the approach is that for most workspaces one cannot define an intrinsic set of corridors (or lanes). In addition, the various turns that have to be generated along a path usually interact; the local planning techniques may result in quite inefficient paths, or they may even fail when a feasible free path exists.

The motion of a point along paths having lower-bounded curvature radius has been investigated in [30], [15], and [22]. Fortune and Wilfong proposed a complete algorithm for deciding whether there exists a feasible path between two configurations among polygonal obstacles [15]. The time and space complexity of this algorithm is exponential in the number of vertices of the obstacles. Using some of the results presented in [15], Jacobs and Canny described an implemented polynomial-time planner [22] which discretizes the boundary of the polygonal obstacles and connects the points resulting from this discretization by paths of standard shapes, called "jumps," which are made of circular arcs and straight segments. The resulting paths may partially lie in contact space. The time complexity of the algorithm is $O((n^3/\delta)\log n + n^2/\delta^2)$, where n is the number of obstacle vertices and δ is the distance between two discretized points in an obstacle edge.

The fact that the results obtained in nonlinear control theory are applicable to characterize the controllability of robots subject to linear equality constraints on the velocity was first pointed out to the Robotics community in [34] and [32]. The main results used in these papers and subsequent ones is the following: If the Lie algebra of the vector fields generated by the controls of a nonholonomic robot has the same dimension as the configuration space of the robot (Controllability Rank Condition), then this robot is controllable. Li and Canny used these results to prove that a ball can reach any configuration on a plane by a pure rolling motion [34]. They also showed that a ball can reach any configuration in contact with a fixed ball by rolling, if the radii of the two balls are different. Laumond and Siméon [32], and Barraquand and Latombe [4], [6], [7], working independently, applied results in nonlinear control to prove that one-body and two-body mobile robots are controllable. However, both proofs consider only the linear equality constraint on the velocity deriving from the rolling contact between the wheels and the ground. Furthermore, they assume that there is no additional inequality constraint imposed on the steering angle by mechanical stops in the steering mechanism. A general strategy for motion planning with nonholonomic constraints using methods of control theory is also presented in [25], but it is currently limited to problems without obstacles.

Our contribution to robot motion planning with nonholonomic constraints reported in this paper is threefold:

1. We show that a careful instantiation of the Controllability Rank Condition Theorem subsumes and generalizes most previous results on the controllability of nonholonomic robots.
2. We apply this result to multibody mobile systems and we derive new formal results related to the controllability of these systems, even in the presence of inequality kinematic constraints.
3. We describe an implemented planner inspired by these results which is effective for both one-body and two-body mobile robots. This planner presents some interesting practical advantages over other planners proposed so far.

3. Nonlinear Control Systems.

Later in this paper we will regard a nonholonomic robot as a nonlinear control system. In this section we recall the definition of some basic concepts in controllability theory and we review important results related to nonlinear control. See [20] for a more detailed presentation of these concepts and [34], [6], and [7] for another presentation of the application of nonlinear control to nonholonomic robots.

3.1. Definition of Controllability. Let Ω be a measurable subset of $\mathbf{R}^m$ and let $\mathscr{C}$ be a connected manifold of dimension n. We consider a control system of the form

$$\dot{q} = f(q, u), \tag{1}$$

where $u \in \Omega$, $q \in \mathscr{C}$, and f is smooth as a function of q. Ω represents the *control space* of the system, i.e., the set of admissible control values. $\mathscr{C}$ represents the *state space*, or *configuration space*, i.e., the set of distinguishable states that the system may take at any given time.

Given a subset $U \subset \mathscr{C}$, the configuration $q_1 \in U$ is said to be *U-accessible* from $q_0 \in U$ if there exists a piecewise constant control $u(t)$, $t \in [t_0, t_1]$, such that the solution $q(t)$ of the system (1) satisfies $q(t_0) = q_0$, $q(t_1) = q_1$ and $q(t) \in U$, for all $t \in [t_0, t_1]$. We write $q_1 A_U q_0$. The set of points U-accessible from q_0 is denoted by $A_U(q_0)$. (*Remark.* In [20], $u(t)$ is more generally defined as a bounded measurable function. This is too general for our multibody mobile robot problems, where u will stand for the pair (linear velocity, steering angle) of the robot's tractor.

Theorems 2 and 3 stated in Section 3.3. are the only results of [20] that we use in this paper. They remain true with this more specific class of control functions.)

The system (1) is *controllable at* q_0 iff $A_{\mathscr{C}}(q_0) = \mathscr{C}$. If this is true for any state $q_0 \in \mathscr{C}$, then the system is *controllable*. This simply means that any state is $\mathscr{C}$-accessible from any other state. However, this global notion of controllability is not easily amenable to a mathematical characterization. In this respect, a more suitable concept is the notion of local controllability defined below.

The system (1) is *locally controllable at* q_0 iff for every neighborhood U of q_0, $A_U(q_0)$ is also a neighborhood of q_0. It is *locally controllable* iff this is true for every $q_0 \in \mathscr{C}$.

Accessibility is a reflexive and transitive relation, but it may not be symmetric. The symmetric closure of this relation is called weak accessibility. q' is *weakly U-accessible* from q iff there exists a sequence $q_0, \ldots, q_r$ such that $q = q_0$, $q' = q_r$, and either $q_i A_U q_{i-1}$ or $q_{i-1} A_U q_i$ for every $i \in [1, r]$. The set of points weakly U-accessible from q is denoted by $WA_U(q)$.

The system (1) is *weakly controllable at* q_0 iff $WA_{\mathscr{C}}(q_0) = \mathscr{C}$. If this is the case, the system is necessarily weakly controllable at any other configuration, because weak accessibility is an equivalence relation; hence, the system is *weakly controllable*. Again, weak controllability is a global concept that has a more useful local equivalent.

The system (1) is *locally weakly controllable at* q_0 if for every neighborhood U of q_0, $WA_U(q_0)$ is also a neighborhood of q_0. It is *locally weakly controllable* if this is true at every $q_0 \in \mathscr{C}$.

Clearly, by patching sequences of open subsets, local controllability implies controllability. In the same way, local weak controllability implies weak controllability. Furthermore, for symmetric systems, i.e., systems for which the accessibility relation is symmetric, (local) weak controllability is equivalent to (local) controllability. Therefore, *for a symmetric system, local weak controllability implies controllability.*

3.2. Frobenius Theorem. Consider the set $X(\mathscr{C})$ of smooth vector fields on $\mathscr{C}$. Using (1), each constant control $u \in \Omega$ defines a vector field $X_u = f(\cdot, u)$ on $\mathscr{C}$. We let F denote the set of all the vector fields corresponding to the admissible values of the control

$$F = \{X \in X(\mathscr{C}) \mid \exists u \in \Omega, X = f(\cdot, u)\}.$$

Let (X, Y) be any pair of vector fields in $X(\mathscr{C})$. Given any configuration $q \in \mathscr{C}$, let us consider a path in $\mathscr{C}$ starting at q and obtained by concatenating the four following paths:

The first path follows the flow of X during δt. (The *integral curve* of a vector field X on $\mathscr{C}$ is a curve whose tangent at every q is $X(q)$. We say that the curve *follows the flow* of X.)

The second path follows the flow of Y during δt.

The third path follows the flow of $-X$ during δt.

The fourth path follows the flow of $-Y$ during δt.

Let q' be the configuration reached at the end of these four paths. A straightforward Taylor expansion shows that

$$\lim_{\delta t \to 0} \frac{q' - q}{\delta t^2} = dY \cdot X - dX \cdot Y,$$

where $dY \cdot X$ and $dX \cdot Y$ denote the products of the $n \times n$ matrices

$$dY = \begin{pmatrix} \frac{\partial Y_1}{\partial q_1} & \cdots & \frac{\partial Y_1}{\partial q_n} \\ \vdots & & \vdots \\ \frac{\partial Y_n}{\partial q_1} & \cdots & \frac{\partial Y_n}{\partial q_n} \end{pmatrix}; \qquad dX = \begin{pmatrix} \frac{\partial X_1}{\partial q_1} & \cdots & \frac{\partial X_1}{\partial q_n} \\ \vdots & & \vdots \\ \frac{\partial X_n}{\partial q_1} & \cdots & \frac{\partial X_n}{\partial q_n} \end{pmatrix},$$

and the n vectors

$$X = (X_1\ X_2 \ldots X_n)^{\mathrm{T}}; \qquad Y = (Y_1\ Y_2 \ldots Y_n)^{\mathrm{T}}.$$

In the above expressions $q_1, \ldots, q_n$ denote the coordinates of q in some chart, and $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ denote the components of the vectors X and Y in the basis of the tangent space $T_q(\mathscr{C})$ induced by this chart. The expression $dY \cdot X - dX \cdot Y$ determines a new vector field which is commonly denoted by $[X, Y]$ and called the *Lie bracket* of X and Y.

By definition, the *Control Lie Algebra* associated with F, denoted by $CLA(F)$, is the smallest subalgebra of $X(\mathscr{C})$ which contains F. Stated otherwise, $CLA(F)$ is the subspace of $X(\mathscr{C})$ generated by all the linear combinations of vector fields in F and all their Lie brackets recursively computed.

For every $q_0 \in \mathscr{C}$, let $CLA(F)(q_0)$ denote the subspace of tangent vectors spanned by the vector fields of $CLA(F)$ at q_0. A connected submanifold $\mathscr{C}'$ of $\mathscr{C}$ is an *integral submanifold* of $CLA(F)$ if at each $q \in \mathscr{C}'$ the tangent space to $\mathscr{C}'$ is contained in $CLA(F)(q)$. $\mathscr{C}'$ is a *maximal integral submanifold* of $CLA(F)$ if it is not properly included in any other integral manifold.

The Frobenius integrability theorem can be stated as follows.

THEOREM 1. *If the dimension of $CLA(F)(q)$ has a constant value k for every $q \in \mathscr{C}$, there exists a partition of $\mathscr{C}$ into maximal integral submanifolds of $CLA(F)$ all of dimension k.*

3.3. Controllability Rank Condition. The system (1) is said to satisfy the *Controllability Rank Condition at q_0* iff the dimension of $CLA(F)(q_0)$ is exactly the dimension n of $\mathscr{C}$. If this is true for every $q_0 \in \mathscr{C}$ then the system is said to satisfy the Controllability Rank Condition.

The following results derive from the work of Chow [11]. They were elucidated in [19], [18], [36], [43], [24], and [20].

THEOREM 2. *If the system* (1) *satisfies the Controllability Rank Condition at* q_0, *then it is locally weakly controllable at* q_0.

THEOREM 3. *If the system* (1) *is locally weakly controllable, then the Controllability Rank Condition is satisfied on an open dense subset of* $\mathscr{C}$.

The two theorems taken together are known as the Controllability Rank Condition Theorem. In particular, if we only consider symmetric systems for which the dimension of $CLA(F)(q)$ does not depend on q, we can infer that a control system is locally controllable (hence, controllable) iff it satisfies the Controllability Rank Condition.

Another presentation of the Controllability Rank Condition Theorem based on the concept of *distribution* can be found in [21]. Its relation to nonholonomic robots is analyzed in [34], [6], and [7]. Unlike this presentation, which only applies to linear equality constraints, the formulation used above allows us to deal with nonlinear equality and inequality constraints, as shown in the next section.

4. Nonholonomic Constraints

4.1. Terminology. We consider a robot $\mathscr{A}$ made of one or several rigid bodies moving in a workspace $\mathscr{W}$. A *configuration* of $\mathscr{A}$ is a specification of the position of every point in $\mathscr{A}$ with respect to a Cartesian frame embedded in $\mathscr{W}$ [37]. The *configuration space* of $\mathscr{A}$ is the space $\mathscr{C}$ of all the possible configurations of $\mathscr{A}$. The configuration space of a mechanical system made of rigid bodies is a smooth manifold [3]. For instance, the configuration space of a two-dimensional rigid body translating and rotating in $\mathscr{W} = \mathbf{R}^2$ is $\mathscr{C} = \mathbf{R}^2 \times S^1$, where S^1 denotes the unit circle. In virtually any practical situation, the range of positions reachable by the robot's bodies can be bounded, making $\mathscr{C}$ into a compact manifold. Let n be the dimension of $\mathscr{C}$. We represent a configuration q as a list $(q_1, \ldots, q_n)$ of n generalized coordinates with appropriate modulo arithmetic on the angular coordinates [26].

Suppose that a scalar constraint of the form

$$F(q, t) = 0, \tag{2}$$

with $q \in \mathscr{C}$ and t denoting time, applies to the motion of $\mathscr{A}$. Assume further that F is smooth with nonzero derivative. Then, in theory, we could use the equation to solve for one of the generalized coordinates in terms of the other coordinates and time. Thus, equation (2) defines a $(n-1)$-dimensional submanifold of $\mathscr{C}$. Therefore, instead of $\mathscr{C}$, which is over-dimensioned, we can choose this submanifold as the actual configuration space of $\mathscr{A}$ and the $n-1$ remaining coordinates as its actual generalized coordinates. Constraint (2) is called a *holonomic* equality constraint [17]. If it depends on t, $\mathscr{A}$'s configuration space is time-dependent, otherwise it is time-independent. (Many usual holonomic constraints, e.g., the prismatic and revolute joints of a manipulator arm, are time-independent.) More generally, there may be k constraints of the form (2). If they are independent, i.e., their Jacobian matrix has full rank, they determine an $(n-k)$-dimensional submanifold of $\mathscr{C}$, which is the actual configuration space of $\mathscr{A}$.

A constraint of the form

$$F(q, t) < 0 \qquad \text{or} \qquad F(q, t) \leq 0$$

typically acts as a mechanical stop or an obstacle. It simply determines a subset of $\mathscr{C}$ having the same dimension as $\mathscr{C}$.

A constraint of the form (2) is only a kinematic constraint of one sort. Another one is a scalar constraint of the form

$$G(q, \dot{q}\ t) = 0 \tag{3}$$

with $\dot{q} \in T_q(\mathscr{C})$, the tangent space of $\mathscr{C}$ at q. The pair $(q, \dot{q})$ belongs to $TB(\mathscr{C})$, the tangent bundle associated with the manifold $\mathscr{C}$. The tangent space represents the space of the velocities of $\mathscr{A}$. The tangent bundle is also called the "phase space" in physics and the "state space" in control theory. The tangent space of a smooth manifold is a vector space having the same dimension as the manifold. Hence, $T_q(\mathscr{C})$ has dimension n for every $q \in \mathscr{C}$. The tangent bundle $TB(\mathscr{C})$ is a smooth manifold of dimension $2n$.

A kinematic constraint of the form (3) is holonomic if it is integrable, i.e., $\dot{q}$ can be eliminated and (3) rewritten in the form (2). Otherwise, the constraint is called a *nonholonomic* equality constraint [17]. As we will see below, a nonholonomic equality constraint restricts the space of velocities achievable by $\mathscr{A}$ at any configuration q to a $(n-1)$-dimensional linear subspace of $T_q(\mathscr{C})$ without affecting the dimension of the configuration space. If there are k independent nonholonomic equality constraints of the form (3), the space of achievable velocities is a subspace of $T_q(\mathscr{C})$ of dimension $n-k$.

A nonholonomic equality constraint is often caused by a rolling contact between two rigid bodies. It expresses the fact that the relative velocity of the two points of contact is zero. When the motion in contact combines rolling and sliding, the expression depends on the friction coefficient of the two bodies, and hence is nonlinear. When there is no sliding, the nonholonomic constraint is linear in $\dot{q}$. The second case, though less general than the first, is much simpler and quite widespread in practice. For instance, in the car example, this corresponds to assuming no slipping of the wheels on the ground.

A constraint of the form

$$G(q, \dot{q}, t) < 0 \qquad \text{or} \qquad G(q, \dot{q}, t) \leq 0$$

is a kinematic inequality constraint. It restricts the set of achievable velocities at any configuration q to a subset of $T_q(\mathscr{C})$, having the same dimension as $T_q(\mathscr{C})$. A constraint bounding the steering angle of a car is a typical kinematic inequality constraint.

When dealing with constraints of the form (3), two important questions arise:

1. *The Integrability question:* Are they nonintegrable? that is, Are we sure that they are actually nonholonomic?
2. *The Controllability question:* Do they restrict the set of configurations reachable from any given configuration?

We investigate these questions in the next two subsections. First, under very general conditions, we show that the concept of kinematic constraint applied to a robot is equivalent to the concept of control system as defined in (1). This equivalence allows us to use results from nonlinear control theory to answer the above questions. Using the Frobenius Integrability Theorem, we give a necessary and sufficient condition of holonomy (and nonholonomy) for equality constraints of the form (3). Then, using the Controllability Rank Condition Theorem we analyze the second question. We state a necessary and sufficient condition under which kinematic constraints, whether they are linear or nonlinear, equality or inequality, have no effect on the range of achievable configurations.

For simplicity, in the rest of the paper, we will assume that the kinematic constraints do not depend on time. However, all the results remain valid when constraints are time-dependent.

4.2. Kinematic Constraints and Control Systems. Let us consider a set of $k < n$ independent kinematic constraints of the form (3)

$$G(q, \dot{q}) = (G^1(q, \dot{q}), \dots, G^k(q, \dot{q})) = (0, \dots, 0).$$

For each q, $G_q = G(q, \cdot)$ defines a function from $T_q(\mathscr{C})$ to $\mathbf{R}^k$. As the k constraints are independent, the Jacobian of this function has full rank. The subset of the tangent space verifying the kinematic constraints is simply $G_q^{-1}(0, \dots, 0)$. According to the Implicit Function Theorem (e.g., see [46], p. 31), this subspace is a submanifold of $T_q(\mathscr{C})$ of dimension $n - k$. We denote a chart (local coordinate system) on this manifold by $u = (u_{k+1}, \dots, u_n)$ and we define $f_q = u^{-1}$. We obtain the following relation:

$$\dot{q} = f_q(u) = f(q, u).$$

Under the additional assumption that f is smooth as a function of q, this relation locally defines a nonlinear control system with $n - k$ controls $(u_{k+1}, \dots, u_n)$. Assume that we impose inequality constraints in addition to the equality constraints mentioned above. These new constraints are transformed into inequality constraints applying to the controls by means of the inverse of the chart u. They define the shape of the set Ω of admissible controls.

Reciprocally, if we consider any control system of the type (1) such that $f(q, \cdot) = f_q$ has full rank as a function of the control $u = (u_{k+1}, \dots, u_n)$, then we can apply again the Implicit Function Theorem for every q and obtain a chart $G_q = (G_q^1, \ldots, G_q^n)$ verifying:

$$\text{for all} \quad \forall i \in [1, k]: \quad G_q^i(f_q(u)) = G_q^i(\dot{q}) = G^i(q, \dot{q}) = 0,$$
$$\forall i \in [k+1, n]: \quad G_q^i(f_q(u)) = G_q^i(\dot{q}) = u_i.$$

The first k equalities precisely define k independent kinematic constraints. Furthermore, the inequalities on the controls that define the shape of the set Ω are transformed into inequalities on the velocity by means of the G_q^i, $i \in [k+1, n]$.

Therefore, in general, a robot subject to a set of k independent kinematic constraints is locally equivalent to a control system with $n - k$ controls for which the function f has full rank in u. Furthermore, any additional inequality constraint on the velocities is equivalent to an inequality constraint on the controls.

4.3. Nonholonomy and Controllability. Consider a robot subject to a set of k independent kinematic equality constraints of the form (3). To answer the integrability question, we first compute the equivalent control system, i.e., the function $f(q, u)$, as indicated above.

We can characterize the integrability of the constraints using the Frobenius Theorem. For each configuration q, the dimension r of $CLA(F)(q)$ is clearly greater than or equal to $n - k$.

If r takes values greater than $n - k$, then, according to Theorem 1 (Frobenius), there exists a partition of $\mathscr{C}$ into maximal integral submanifolds of dimension greater than $n - k$. Hence, the constraints are nonintegrable. Indeed, if they were, there would be a single maximal integral manifold of dimension $n - k$ (that we could choose as the actual configuration space of $\mathscr{C}$). Hence, the constraints are nonholonomic.

On the other hand, if r is equal to $n - k$ at every q, then the Frobenius Theorem entails that the maximal integral manifolds of $CLA(F)$ have dimension $n - k$. Therefore, the admissible configurations of the robot span an $(n - k)$-dimensional submanifold of $\mathscr{C}$. As a consequence, the velocities always belong to the tangent space of this submanifold, which is precisely $CLA(F)(q)$. But these same velocities also belong to an $(n - k)$-dimensional submanifold S of the tangent space of $\mathscr{C}$ at q defined by the constraints. Therefore, S is necessarily equal to $CLA(F)(q)$, hence linear. This implies that the k equality constraints are *linear* in $\dot{q}$. This result, though intuitively clear, is worth being emphasized. To characterize holonomic constraints, we can therefore limit ourselves to those which are linear in the velocity parameters. In that case, we can replace the configuration space $\mathscr{C}$ by the maximal integral submanifold passing through the initial configuration of the robot, and get rid of the constraints. The equations defining this submanifold can be written locally in the form (2), i.e., $F(q) = 0$, which is the integral form of the constraints (3). By differentiating this last equation as a function of time, we find again the constraints on the velocity

$$dF(q) \cdot \dot{q} = 0,$$

which gives a more intuitive explanation of the fact that holonomic constraints are necessarily linear in the velocity parameters.

In summary:

PROPOSITION 1. *Kinematic constraints that are properly nonlinear as functions of the velocity are necessarily nonholonomic.*

PROPOSITION 2 (Characterization of Holonomy). *A robot subject to k independent equality constraints of the form* (3) *is holonomic if and only if the codimension $n - r$ of the Control Lie Algebra is equal to the number k of constraints. In such a case, the kinematic constraints are necessarily linear in the velocity parameters.*

The answer to the controllability question for robots subject to kinematic constraints is a direct consequence of the Controllability Rank Condition Theorem. As outlined above, given k independent constraints, we consider the equivalent control system with $n - k$ controls. Then we analyze the dimension r of the Control Lie Algebra by recursively computing the Lie brackets of constant control vector fields. If this number r is constant and equal to n, then the system (i.e., the robot) is locally weakly controllable. Reciprocally, if the system (i.e., the robot) is locally weakly controllable, then r is equal to n on an open dense subset of $\mathscr{C}$. If r varies on $\mathscr{C}$, complex phenomena may occur. The study of these phenomena forms the basis for the so-called Catastrophe Theory [41].

PROPOSITION 3 (Characterization of Controllability). *A robot subject to kinematic constraints on the velocity—which may be linear or nonlinear, equality or inequality—is locally weakly controllable if the dimension r of the Control Lie Algebra is maximal, i.e., equal to the dimension n of the configuration space.*

5. Application to Multibody Mobile Robots. Let us first consider a front-wheel-drive four-wheel car (one-body mobile robot). Our presentation can easily be modified to treat other types of car-like robots. We model the car as a two-dimensional object translating and rotating in the plane (Figure 2). The configuration space of the car is $D \times S^1$, where D is a compact domain of $\mathbf{R}^2$. We parametrize the car configuration by the coordinates X_1 and Y_1 of the midpoint P_1 between the two rear wheels and the angle θ_1 between the x-axis of the Cartesian frame embedded in the plane and the main axis of the car. The velocity parameters are $\dot{X}_1$, $\dot{Y}_1$, and $\dot{\theta}$. The *control parameters* of the car are the *velocity* $v \in \mathbf{R}$ of the midpoint P_0 between the two front wheels (if $v > 0$, the car moves forward) and the *steering angle* φ measuring the orientation of the velocity of P_0 with respect to the main axis of the car (if $0 < \varphi < \pi$ and $v > 0$ the car turns to the left).

Assume that the contacts between the wheels and the ground are pure rolling contacts between two rigid bodies. Hence, in particular, there is no slipping.

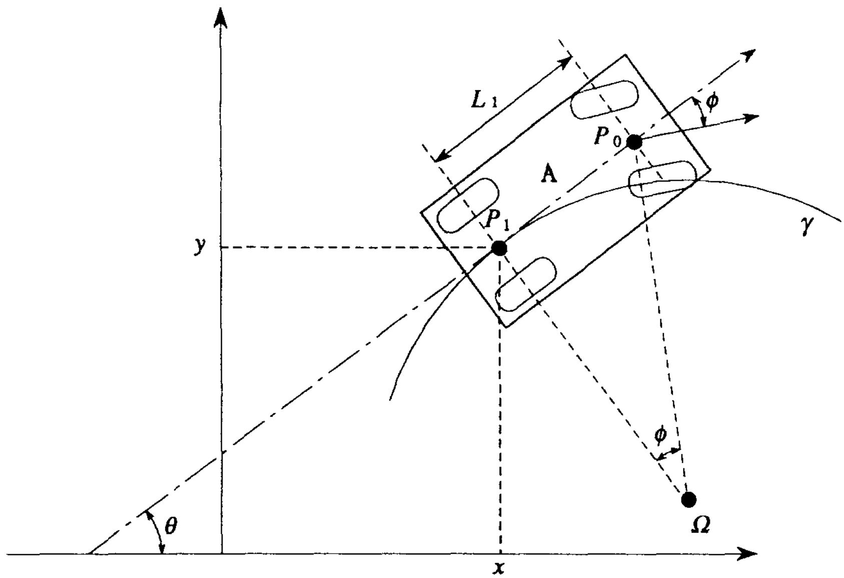

Fig. 2. Car-like robot.

Therefore, the velocity of P_1 points (positively or negatively) along the main axis of the car. We have

$$\dot{X}_1 = \lambda \cos \theta_1, \qquad \dot{Y}_1 = \lambda \sin \theta_1.$$

The elimination of λ yields the following kinematic constraint on the velocity

$$-\dot{X}_1 \sin \theta_1 + \dot{Y}_1 \cos \theta_1 = 0. \tag{4}$$

The equivalent control system is easily computed

$$\left.\begin{aligned} \dot{X} &= v \cos \varphi \cos \theta_1, \\ \dot{Y}_1 &= v \cos \varphi \sin \theta_1, \\ L_1 \dot{\theta}_1 &= v \sin \varphi. \end{aligned}\right\} \tag{5}$$

As v can take both positive and negative values, the system is *symmetric*. In such a case, weak controllability is equivalent to controllability (see Section 3.1), and the results of Section 4 are applicable.

In the two-body mobile robot example of Figure 1, under the same assumption that the contacts between the wheels and the ground are pure rolling contacts between rigid bodies, there are two kinematic constraints. The velocity of the midpoint between the rear wheels of each body is tangent to the orientation of the body. More generally, we can consider a p-body mobile robot consisting of a tractor towing $p-1$ trailers sequentially hooked. The midpoint between the rear wheels of the first body (the tractor) is denoted by P_1. The midpoint between the rear wheels of the kth body is denoted by P_k. We therefore have p points $P_1, \ldots, P_p$, whose coordinates are denoted by $(X_1, Y_1), \ldots, (X_p, Y_p)$. The orientation of the kth body with respect to the x-axis of the Cartesian frame embedded in the plane is denoted by θ_k. The configuration space of the p-body robot is $D \times (S^1)^p$, where D is a compact domain of $\mathbf{R}^2$. Its dimension is $n = p + 2$. We parametrize the configuration by $(X_1, Y_1, \theta_1, \ldots, \theta_p)$. The velocity parameters are $\dot{X}_1$, $\dot{Y}_1$, $\dot{\theta}_1, \ldots, \dot{\theta}_p$. The control parameters are the same as for the car, that is, the velocity v and the steering angle φ.

There are p kinematic equality constraints, one for each body. To establish these constraints, it is convenient to represent the points $P_1, \ldots, P_p$ in the complex plane, i.e., $P_k = X_k + iY_k$. L_k denoting the length of the kth body, we can write the geometric constraint between the bodies $k-1$ and k as

$$P_k = P_{k-1} - L_k \exp(i\theta_k),$$

which can be rewritten

$$P_k = P_1 - \sum_{l=2}^{k} L_l \exp(i\theta_l). \tag{6}$$

The kinematic constraint of the kth body is

$$\dot{P}_k = \lambda_k \exp(i\theta_k),$$

which is equivalent to

$$\Im(\exp(-i\theta_k)\dot{P}_k) = 0,$$

where $\Im(z)$ denotes the imaginary part of the complex number z. Combining this characterization with the derivative of (6) and using the linearity of the $\Im$ operator, we obtain the kth kinematic constraint

$$-\dot{X}_1 \sin\theta_k + \dot{Y}_1 \cos\theta_k - \sum_{l=2}^{k} L_l \dot{\theta}_l \cos(\theta_l - \theta_k) = 0.$$

In particular, for $k = 1$, we obtain

$$-\dot{X}_1 \sin\theta_1 + \dot{Y}_1 \cos\theta_1 = 0, \tag{7}$$

which is precisely the kinematic constraint (4) of the car-like robot.

For $k = 2$, we get

$$-\dot{X}_1 \sin\theta_2 + \dot{Y}_1 \cos\theta_2 - L_2\dot{\theta}_2 = 0. \tag{8}$$

Equations (7) and (8) are the kinematic constraints of the two-body robot problem.

Similarly, by combining $\dot{P}_k = \lambda_k \exp(i\theta_k)$ with the derivative of

$$|P_k - P_{k-1}|^2 = L_k^2$$

we get

$$\lambda_k = \cos(\theta_k - \theta_{k-1})\lambda_{k-1},$$

and by induction

$$\dot{P}_k = v \cos \varphi \left(\prod_{l=2}^{k} \cos(\theta_l - \theta_{l-1}) \right) \exp(i\theta_k).$$

Hence, the equivalent control system of a multibody mobile robot is composed of the equations (5) and

$$L_k \dot{\theta}_k = v \cos \varphi \left(\prod_{l=2}^{k-1} \cos(\theta_l - \theta_{l-1}) \right) \sin(\theta_{k-1} - \theta_k), \qquad \text{for all} \quad k \in [2, p].$$

Let $X(v, \varphi)$ denote the vector field corresponding to the constant control (v, φ). We get

$$X(v, \varphi) = \cos(\varphi)X(v, 0) + \sin(\varphi)X(v, \pi/2).$$

Let us take any two fields corresponding to two different values of the steering angle φ_1 and φ_2. The Control Lie Algebra generated by $X(v, \varphi_1)$ and $X(v, \varphi_2)$ is the same as the one generated by $X(v, 0)$ and $X(v, \pi/2)$, because of the bilinearity of the Lie Bracket operation

$$[X(v, \varphi_1), X(v, \varphi_2)] = \sin(\varphi_1 - \varphi_2)[X(v, 0), X(v, \pi/2)].$$

Therefore, the dimension of the Control Lie Algebra of the multibody car system is not affected by inequality constraints on the steering angle. It has been shown in [6] and [7] that this dimension is maximal for one-body and two-body mobile robots. Below we extend this result to a three-body robot.

The constant control vector field space is generated by the two following vector fields

$$X_1 = X(1,0) = \left(\cos\theta_1, \ \sin\theta_1, \ 0, \ \frac{\sin(\theta_1 - \theta_2)}{L_2}, \ \cos(\theta_1 - \theta_2)\frac{\sin(\theta_2 - \theta_3)}{L_3} \right),$$

$$X_2 = L_1 X(1, \pi/2) = (0, \ 0, \ 1, \ 0, \ 0),$$

whose Lie bracket is

$$X_3 = [X_1, X_2] = \left(-\sin\theta_1, \cos\theta_1, 0, \frac{\cos(\theta_1 - \theta_2)}{L_2}, -\frac{\sin(\theta_1 - \theta_2)\sin(\theta_2 - \theta_3)}{L_3} \right).$$

We next compute

$$X_4 = L_2[X_1, X_3] = \left(0, 0, 0 \frac{1}{L_2}, -\frac{\cos(\theta_2 - \theta_3)}{L_3} \right)$$

and

$$X_5 = \cos(\theta_1 - \theta_2)[X_1, X_4] - \sin(\theta_1 - \theta_2)[X_3, X_4]$$
$$= \left(0, 0, 0, \frac{1}{L_2^2}, -\frac{1}{L_3^2} - \frac{\cos(\theta_2 - \theta_3)}{L_2 L_3}\right).$$

Finally

$$|\det(X_1, X_2, X_3, X_4, X_5)| = \frac{1}{L_2 L_3^2} \neq 0.$$

Using a proof by recurrence, Laumond extended the above result previously published in [8] to a multibody mobile robot with an arbitrary number of trailers, hence showing that the Control Lie Algebra of a p-body mobile robot has dimension $p+2$ [33].

Wrapping up all the results presented above, we can state:

PROPOSITION 4. *A multibody mobile robot is controllable whenever the steering angle φ of the tractor can take at least two different values φ_1 and φ_2 in $(-\pi, +\pi]$ such that $|\varphi_2 - \varphi_1| \neq \pi$.*

In particular:

A multibody mobile robot is controllable if the steering angle φ is constrained to take values in a subinterval of $[-\pi, +\pi]$ of nonzero length, e.g., $\varphi \in [-\varphi_{max}, +\varphi_{max}]$ with $0 < \varphi_{max} < \pi/2$ (the most usual case in practice).

A multibody mobile robot that can only turn to the left, i.e., such that $\varphi \in \varphi_{min}, -\varphi_{max}] \subset [0, \pi]$ is controllable, i.e., is "maneuverable to the right" (see Figure 7). This kind of constraint applies to some cheap remote controlled car.

If there exists a feasible free path for a multibody mobile robot with limited steering angle $\varphi \in [\varphi_1, \varphi_2] \subset [-\pi, +\pi]$, then there exists a feasible free path that uses only the extremal values of the steering angle.

All these statements are direct consequences of the fact that the dimension of the Control Lie Algebra is not affected by the choice of the steering angles.

6. Planning with Nonholonomic Constraints. We now describe an implemented planner deriving from the mathematical results presented above. In theory, the method used by the planner is applicable to any multibody mobile robots whose steering angle φ takes values in $[\varphi_{min}, \varphi_{max}] \subset (-\pi/2, +\pi/2)$. The planner is approximate in the sense that, if it generates a path, this path is only guaranteed to end in a prespecified neighborhood of the goal configuration. This neighborhood can be set as small as we wish.

Following the description of the planner, we establish a claim saying that the algorithm is *asymptotically complete*, i.e., for any given problem that admits a solution path, the planner is guaranteed to generate a solution path, provided that the discretization of the search parameters has been set fine enough. We also establish two claims saying that the planner is *asymptotically optimal*, i.e., for any given problem that admits a solution path, the planner is guaranteed to generate a solution path with minimal number of reversals (changes of sign of the linear velocity), provided that the discretization of the search parameters has been set fine enough. The first optimality claim applies to car-like robots only, while the second, which is slightly weaker, applies to robots with more than one body. These claims are not constructive, but in addition to characterizing the asymptotic behavior of the planner, they can be particularly useful when robot control is imperfect and a prior estimate of navigation accuracy is available.

Finally, we show experimental results obtained with the planner. The time/space complexity of the planner is exponential in the number of bodies of the mobile robot. This currently limits the practicality of our implementation to robots with one and two bodies. Our experiments have been conducted in these two cases only.

The restriction that the steering angle be strictly comprised between $-\pi/2$ and $+\pi/2$ is only aimed at avoiding the cases where the robot rotates with a zero linear velocity (hence, with a zero turning radius). Including $\pm\pi/2$ in the range of values of φ would make the notion of reversal unclear. The planner could certainly be adapted, but the claim that it is asymptotically optimal would then become meaningless.

6.1. Description of the Planner. Let the workspace $\mathscr{W}$ of a multibody mobile robot $\mathscr{A}$ be populated by stationary obstacles $\mathscr{B}_i$, $i = 1, \ldots, q$. These obstacles map in the configuration space $\mathscr{C}$ of $\mathscr{A}$ to regions $\mathscr{C}\mathscr{B}_i$ called *C-obstacles* and defined by

$$\mathscr{C}\mathscr{B}_i = \{q \in \mathscr{C} \mid \mathscr{A}(q) \cap \mathscr{B}_i \neq \emptyset\},$$

where $\mathscr{A}(q)$ denotes the region of $\mathscr{W}$ occupied by $\mathscr{A}$ at configuration q. The subset $\mathscr{C}_{\text{free}} = \mathscr{C} \setminus \bigcup_{i=1}^{q} \mathscr{C}\mathscr{B}_i$ is called the *free space*. We model both $\mathscr{A}$ and the $\mathscr{B}_i$'s as closed regions. Therefore, $\mathscr{C}_{\text{free}}$ is an open subset of $\mathscr{C}$, hence a manifold of dimension n [26].

Given two configurations q_1 and q_2 in $\mathscr{C}_{\text{free}}$, the path planning problem is to construct a path connecting q_1 to q_2 and lying in $\mathscr{C}_{\text{free}}$, i.e., a continuous map

$$\tau: s \in [0, 1] \mapsto \tau(s) \in \mathscr{C}_{\text{free}}$$

such that $\tau(0) = q_1$ and $\tau(1) = q_2$. In addition, we impose that τ be of class piecewise C^1 and that the tangent to this path, $d\tau/ds$, wherever it is defined, lie in the subset of the tangent space of $\mathscr{C}$ selected by the kinematic constraints.

Our planner assumes that the steering angle φ takes its values in

$$[\varphi_{\min}, \varphi_{\max}] \subset (-\pi/2, +\pi/2).$$

A planning problem is defined by the geometry of the robot and the workspace, the initial and goal configurations (q_1 and q_2), a neighborhood $\mathscr{G}(q_2)$ of the goal configuration (to be explained later), and the two extremal steering angles ($\varphi_{\min}$ and $\varphi_{\max}$). Three tuning parameters, δt_0, R, and H, are also given to the planner (see below). The planner generates a path by exploring a discrete subset of $\mathscr{C}_{\text{free}}$. This exploration consists of concurrently constructing and searching a tree T. The root of the tree is the initial configuration q_1. At any time during the search, every other node of the current T is a configuration already attained by the search. A list *OPEN* contains all the leaves of the current T whose successors have not been generated yet. The search iteratively selects a configuration q in *OPEN*, removes it from *OPEN*, and computes four successors of q by setting the two control parameters v and φ to the four values in

$$\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\},$$

and integrating the velocity parameters $\dot{X}_1$, $\dot{Y}_1$, $\dot{\theta}_1, \ldots, \dot{\theta}_p$ of the robot over the given constant interval of time δt_0 using the differential equations established in the previous section. The choice of δt_0 (an input to the planner) determines the grain of the discretization. The three equations of a one-body robot can easily be integrated analytically. For the fourth equation of a two-body robot our planner uses a fourth-order Runge–Kutta method. A computed successor q' of q is inserted in T as a child of q iff (1) the path from q to q' is collision-free, and (2) q' is not "too close" from a configuration already in T. The implementation of these two conditions is described below.

The implemented planner only verifies that q' is collision-free (not the entire path connecting q to q'). This is done by intersecting the robot at q' with the obstacles using a simple divide-and-conquer technique described in [4]. The workspace is represented as a bitmap so that the obstacles can have arbitrary shape without affecting the time complexity of the collision-checking algorithm. The test, however, is not completely safe. This problem can be eliminated by precomputing the maximal Euclidean displacement ρ of the points in $\mathscr{A}$ during any of the four incremental motion steps defined above and isotropically growing the obstacles in the workspace bitmap by ρ. Then the collision-checking test becomes conservative. If the robot is a one-body car and both the robot and the obstacles are polygonal, then an even better approach is to check collisions exactly. By taking advantage of the fact that the incremental motion of every vertex of the robot (resp. the obstacles) relative to the obstacles (resp. the robot) is a circular arc (or a line segment if $\varphi_{\min}$ or $\varphi_{\max}$ is null), the computation can easily be done in time $O(ab)$, where a and b are the number of vertices in the robot and the obstacles, respectively [44].

We could check that q' is not too close from a configuration already in T by defining a metric in the configuration space $\mathscr{C}$ and verifying that the minimal distance between q' and the configurations currently in T is greater than some prespecified threshold. Instead, to simplify the test and make it as fast as possible, we represent the configuration space $\mathscr{C} = \mathbf{R}^2 \times (S^1)^p$ as a $(p + 2)$-dimensional hyperparallelepiped $[X_1^{\min}, X_1^{\max}] \times [Y_1^{\min}, Y_1^{\max}] \times [0, 2\pi)^p$ that we decompose into an array A of $2^{R(p+2)}$ smaller parallelepipeds of equal size, called *cells*. The parameter R is the resolution of the array. A cell is said to be explored if it contains a configuration in T. A new configuration q' is inserted in T if it does not belong to an explored cell. Hence, the array A is used as a simple device for indexing the configurations attained by the search algorithm. Since the search accepts at most one configuration per cell, the resolution of A should be fine enough so that none of the successors of any configuration q lie in the same cell as q. Therefore, the choices of δt_0 and R are not totally independent.

The planner explicitly represents A as an array of bits. Hence, checking whether a new configuration q' is in an explored cell takes constant time. Since collision checking requires more time, the planner first verifies that a newly generated configuration q' is not in an explored cell; then, only if this is necessary, it performs the collision-checking test.

The tree T is constructed as it is searched by a Dijkstra algorithm [1]. At every iteration, it selects a configuration q in *OPEN* that has been achieved with minimal number of reversals. In addition, it cuts the search at depth H (an input to the planner). This means that, if the configuration q selected in *OPEN* is at depth H in the search tree, no successor is generated; the planner removes q from *OPEN* and immediately selects another configuration. Hence, the maximal number of nodes in the search tree is bounded, so that the search always terminates in a finite amount of time. It terminates with success, when it selects a configuration in *OPEN* that lies in the given goal neighborhood $\mathscr{G}(q_2)$, or with failure, when it starts a new iteration with an empty *OPEN* list. Thus, the planner may return a path that does not attain the goal configuration exactly.

The asymptotic worst-case time required by the planner is proportional to

$$K \times (S + C),$$

where K is the maximal size of the search tree, S is the time necessary to select the best configuration in *OPEN* and insert its accepted successors in *OPEN*, and C is the time necessary to check that a new configuration is collision-free. We have $K \leq \min\{4^H, |A|\}$, where $|A|$ denotes the size of A. Let us assume that $|A| < 4^H$ and $|A| = 2^{nR}$ ($n = p + 2$ is the dimension of the configuration space). We represent *OPEN* as a heap so that S is logarithmic in the maximal size of *OPEN*, which is of the same order of magnitude as the size of A. Let 2^{2R} be the size of the workspace bitmap in which the collision-checking operation is performed. Our simplified collision-checking algorithm, which in the worst-case "draws" the full contour of the robot in the workspace bitmap, takes $O(2^R)$ time. Hence, posing $c = 2^R$ (the number of discretization intervals along each dimension of the configuration space), the time complexity of the planner is $O(c^n(n \log c + c))$ and the space complexity $O(c^n)$. Both are exponential in the dimension n of the configuration space, hence in the number p of bodies of the robot.

6.2. Asymptotic Completeness and Optimality. The following three claims are established assuming a perfect collision-checking operation.

Claim 1 (Asymptotic Completeness). *If q_1 and q_2 are contained in the same connected component of $\mathscr{C}_{\text{free}}$, then the planner will find a path connecting q_1 to a* *configuration in the goal neighborhood $\mathscr{G}(q_2)$, provided that we have set δt_0 small enough, H large enough, and R large enough.*

PROOF. This claim derives from Proposition 4, with $\varphi_1 = \varphi_{\min}$ and $\varphi_2 = \varphi_{\max}$. In more explicit terms, the proposition tells us that if two configurations q_1 and q_2 lie in the same connected component of $\mathscr{C}_{\text{free}}$ (an open set), then there exists a finite sequence of controls $u_1, \ldots, u_r \in \{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$, with respective finite durations $\delta t_1, \ldots, \delta t_r$, such that applying these controls successively with their specified durations produces a path that connects q_1 to q_2 in $\mathscr{C}_{\text{free}}$.

Let us consider a path τ (any one) defined as above. Assume some metric in $\mathscr{C}$. Let $\eta > 0$ be the smallest distance between τ and the C-obstacles. For any ε such that $0 < \varepsilon < \eta$, if δt_0 is chosen small enough, then there exists a finite sequence of controls in $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$, each applied with the *same* duration δt_0 (two successive controls may be identical), that moves the robot along a path τ_ε that remains closer to τ than ε and whose endpoint $\tau_\varepsilon(1)$ is closer to $\tau(1) = q_2$ than ε (see Appendix A).

Let T_∞ be the infinite tree implicitly defined by the initial configuration q_1 (the root of T_∞), all its successors accessible by collision-free paths, each obtained by applying one of the four controls used by the planner during δt_0 (here, we use no indexing array), and all the successors of these successors recursively computed. (If the same configuration is attained by several different paths in the tree, it is represented as multiple nodes.) For a sufficiently small value of δt_0, T_∞ is guaranteed to contain at least one path τ_ε defined as above. Let N be the minimal length of the sequence of controls producing a solution path contained in T_∞. (Such a path does not necessarily satisfy the above definition of a path τ_ε.)

We define T_H as the finite subtree of T_∞ made of the first $H+1$ layers of T_∞, including q_1 (as the first layer). Let $H \geq N$, so that T_H contains at least one solution path. T_H is further reduced by interrupting each of its paths beyond the first node representing a configuration contained in $\mathscr{G}(q_2)$, if such a node exists along the path. We can set the resolution R of A large enough so that T_H contains a solution path τ_{sol} such that every node of T_H along this solution path represents a configuration q lying in a cell of A that q shares with no other distinct configuration q' of T_H, except if the node representing q' is in another solution path that satisfies the same property as τ_{sol}, or if the path in T_H connecting q_1 to the parent of q' contains more reversals than the subpath of τ_{sol} connecting q_1 to the parent of q. Then the search algorithm of the planner is guaranteed to find a path to a configuration contained in the goal neighborhood $\mathscr{G}(q_2)$. Indeed, since the size of the search tree is bounded, the search is guaranteed to terminate, and it cannot terminate with failure since this tree necessarily contains at least one solution path whose discovery cannot be prevented by the marking of the indexing array A.

It now remains to verify that the relationships between δt_0, R, and H are not inconsistent. First, δt_0 should be small enough so that there exists a solution path in T_∞. An admissible value of δt_0 determines:

(1) a minimal value $h_{\min}$ for H (the minimal number of motion increments in a solution path); and
(2) a minimal value $r_{\min}^{(1)}$ for R (so that a motion of duration δt_0 is guaranteed to move out of a cell).

In turn, the minimal value of H, $h_{\min}$, determines a minimal value $r^{(2)}_{\min}$ of R (based on the discussion in the above paragraph). Hence, it is possible to appropriately set δt_0, H, and R, in that order. □

The above proof is not constructive. Hence, if a problem admits a solution, we do not know in advance how we should set δt_0, H, and R to guarantee that a solution path will be found. Hence, for any problem, when the planner returns failure, it may be that the problem has no solution, or that we have incorrectly set δt_0, H, and/or R. However, if we have some prior estimate of the precision of the robot navigation system, we may estimate δt_0 according to this knowledge, and, from there, H and R. A failure by the planner then typically corresponds to the case where there is no obvious path that the robot could perform safely without needing further sensory interaction.

CLAIM 2 (Asymptotic Optimality for One-Body Robot). *Let q_1 and q_2 be two configurations in the same connected component of the free space of a car-like robot. Let λ be the minimal number of reversals in a solution path connecting q_1 to q_2, over all possible solution paths. Then the planner will find a path with λ reversals connecting q_1 to a configuration in the goal neighborhood $\mathscr{G}(q_2)$, provided that we have set δt_0 small enough, H large enough, and R large enough.*

(Actually, this claim holds for a slightly modified planner, as described in the following proof.)

PROOF. Consider a one-body robot (i.e., a car). Property 5 (established in Appendix B) tells us that there exists a path τ obtained by applying a finite sequence of controls $u_1, \ldots, u_r \in \{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$, with respective finite durations $\delta t_1, \ldots, \delta t_r$, which connects q_1 to q_2 with λ reversals. Like in the proof of Claim 1, we can approximate τ by a path τ_ε produced by applying a finite sequence of controls in $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$, each applied with the same duration δt_0. By choosing δt_0 small enough, τ_ε remains closer to τ than ε (hence, lies in $\mathscr{C}_{\text{free}}$) and the endpoint $\tau_\varepsilon(1)$ is closer to $\tau(1) = q_2$ than ε (hence, is in the goal neighborhood). In addition, τ_ε has the same number of reversals as τ (see Appendix A).

We define T_∞ and T_H as in the proof of Claim 1. Let N be the minimal length of the sequence of controls producing a solution path with λ reversals contained in T_∞. Let $H \geq N$, so that T_H contains at least one solution path with λ reversals. We can set the resolution of A large enough so that T_H contains a solution path τ_{opt} with λ reversals such that every node of T_H along this solution path represents a configuration q lying in a cell of A that q shares with no other distinct configuration q' of T_H, except if the node representing q' is in another solution path with λ reversals that satisfies the same property as τ_{opt}, or if the path in T_H connecting q_1 to the parent of q' contains more reversals than the subpath of τ_{opt} connecting q_1 to the parent of q.

Let us assume (just for a moment) that all the configurations in T_H are distinct. Then the search algorithm of the planner (which uses the number of reversals as the cost function) is guaranteed to find a path with λ reversals to a configuration contained in the goal neighborhood $\mathscr{G}(q_2)$. Indeed, since the size of the search tree is bounded, the search is guaranteed to terminate; it cannot terminate with failure since this tree necessarily contains at least one solution path whose discovery cannot be prevented by the marking of A; and it will not return a path with more than λ reversals since this tree necessarily contains at least one solution path with λ reversals whose discovery cannot be prevented by the marking of A.

Now let T_H contain several times the same configuration. We should distinguish two cases. The first case is when various occurrences of some configuration are in the same path of the tree. It happens when the robot returns back to a previously attained configuration. The use of the indexing array A then prevents the search algorithm to loop. The second case is when two occurrences of the same configuration (call it q) are in different paths of T_H. This case may affect the optimality of the path. Call $q^{(1)}$ and $q^{(2)}$ the two nodes of T_H representing these two occurrences of q. Assume that T_H contains a path with λ reversals that traverses $q^{(1)}$. If the configuration $q^{(1)}$ is attained with some linear velocity (say, $+1$) and the configuration $q^{(2)}$ with the other linear velocity (i.e., -1), then the best path passing through $q^{(2)}$ contains $\lambda+1$ reversals and is not optimal. If the search algorithm attains $q^{(2)}$ before $q^{(1)}$ (a case that we cannot eliminate), then the use of the indexing array A prevents the planner to later accept $q^{(1)}$ in the search tree and, consequently, to find the optimal path passing through $q^{(1)}$. If this path is the unique optimal path in T_H, or if all the other optimal paths are prevented to be found in the same fashion, then the planner does not return the optimal path.

This problem occurs only if T_H contains two occurrences of the same configuration in two different paths, with the two occurrences attained with different linear velocities. (The case of two occurrences attained with the same linear velocity raises no problem.) One easy way to solve this problem is to slightly modify the algorithm of the planner as follows. Rather than using one indexing array A, the new algorithm uses *two* arrays $A_{(-)}$ and $A_{(+)}$. All the cells in both arrays are initially marked "unexplored." If the search attains a configuration q, the corresponding cell in $A_{(-)}$ (if q is achieved with the linear velocity -1) or in $A_{(+)}$ (if q is achieved with the linear velocity $+1$), is marked "explored." A successor q' of a configuration q attained with the linear velocity -1 (resp. $+1$) is included in the search tree T only if the corresponding cell in $A_{(-)}$ (resp. $A_{(+)}$) is marked "unexplored" (and q' passes the collision-checking test). This modification does not change the order of magnitude of the time/space complexity of the planner, nor the validity of Claim 1. □

The need for two indexing arrays suggested in this proof is illustrated by the example shown in Figure 3. The black region depicts obstacles. In order to go from configuration q_1 to configuration q_2 the car has two possible routes that both pass through configuration q. The two paths differ between q_1 and q; one, call it path 1, passes above the circular obstacle on the left, while the other, path 2, passes below the obstacle. The two paths coincide between q and q_2. Path 1 contains no reversal; path 2 contains one. If the planner uses a single indexing array A, and attains q through path 2 first, then it cannot generate path 1 to

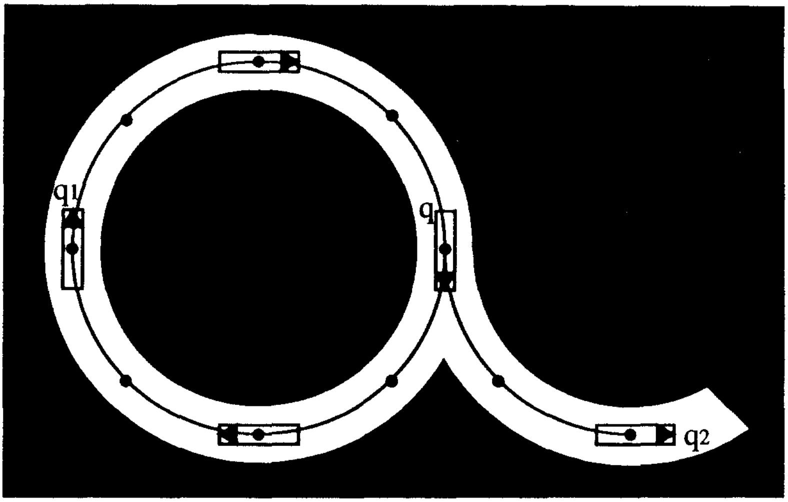

Fig. 3. Need for two indexing arrays.

connect q_1 to q. In contrast, if it uses two arrays $A_{(-)}$ and $A_{(+)}$, it can still generate path 1, and will ultimately return it.

Like the proof of Claim 1, the above proof is not constructive. Hence, although we know that the planner is asymptotically complete and optimal, for a given problem involving a car-like robot, we do not know how to set δt_0, H, and R, in order to guarantee that the planner will return a path with minimal number of reversals if there exists a solution path. Again, we can set δt_0, H, and R according to our estimate of the robot precision.

The above proof would remain valid for multibody robots, if we could extend the proof of Proposition 5 to these robots (see Appendix B). In any case, it can easily be modified in order to derive the following claim for multibody mobile robots.

CLAIM 3 (Asymptotic Optimality for Multibody Robot). *Let q_1 and q_2 be two configurations in the same connected component of the free space of a multibody mobile robot, and let λ be the minimal number of reversals in a solution path connecting q_1 to q_2, over all possible solution paths. Let us modify the planner so that, at every iteration, the search algorithm computes 2r successors of the configuration q selected in OPEN, each generated with a control in* $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\min} + \delta\varphi, \ldots, \varphi_{\min} + (r-1)\delta\varphi\}$, *with* $\delta\varphi = (\varphi_{\max} - \varphi_{\min})/(r-1)$, *where* $r > 1$ *is a discretization parameter given to the planner. Then the planner will find a path with λ reversals connecting q_1 to a configuration in the goal neighborhood $\mathscr{G}(q_2)$, provided that we have set δt_0 small enough, H large enough, R large enough, and r large enough.*

In order to reuse the proof of Claim 2, we should notice that any path τ_{opt} of the multibody robot with minimal number of reversals λ can be approximated as close as we wish by a path with the same number of reversals that is generated by applying a finite sequence of controls in

$$\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\min} + \delta\varphi, \ldots, \varphi_{\min} + (r-1)\delta\varphi\} \in (\varphi_{\min}, \varphi_{\max}),$$

with $\delta\varphi = (\varphi_{\max} - \varphi_{\min})/(r-1)$, each during the same interval of time, by choosing r large enough. Indeed, we can construct a sequence $\{u_i\}_{i=1,2,\ldots}$ of piecewise constant control functions, each with values in $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\min} + \delta_i\varphi, \ldots, \varphi_{\min} + (r_i - 1)\delta_i\varphi\} \in (\varphi_{\min}, \varphi_{\max})$, with $\delta_i\varphi = (\varphi_{\max} - \varphi_{\min})/(r_i - 1)$, each value being applied over the same interval of time, that converges toward the control function generating τ_{opt}. The equations of motion of Section 5 show that the derivatives of the configuration parameters, $\dot{X}_1, \dot{Y}_1, \dot{\theta}_1, \ldots, \dot{\theta}_p$, are continuous functions of the control $u = (v, \varphi)$. Hence, the path τ_i obtained by integrating these equations (over a compact interval) with the control function u_i can be made as close as we want to τ_{opt} by choosing i large enough.

6.3. Experimental Results. The planner described above has been implemented as a program written in C and running on a DEC 3100 MIPS-based workstation. Despite its conceptual simplicity, the planner solves tricky planning problems in reasonable amounts of time, as illustrated by the following experimental results.

In all the examples shown below, the workspace is input as a 2^{2R} bitmap (with $R = 7$, 8, or 9, depending on the examples). The planner uses a single indexing array A of size 2^{nR} (with the same value of R as for the workspace bitmap). The value of δt_0 is set such that each motion step is of the same order of magnitude (slightly greater) than the L^1 diameter of a cell of A. A path planner such as ours is typically used over a workspace whose diameter is ten to thirty times the main dimension of the robot. The values of R and δt_0 used in our experiments are consistent with the precision of most existing mobile robots. The value of H is arbitrarily set to the number of cells in A.

The implemented algorithm producing the experimental results shown below differs slightly from that described in Sections 6.1 and 6.2. First, the search terminates when it attains a configuration lying in the cell of A that contains the goal configuration. Second, as mentioned above, it uses a single indexing array A (as in Section 6.1). Third, whenever the range of values $[\varphi_{\min}, \varphi_{\max}]$ of the steering angle φ contains 0, this value is included in the discretized set of controls used by the planner. Hence, at each iteration, the search algorithm computes six successors (instead of four) of the configuration selected in *OPEN*. Among the configurations that tie in *OPEN* (i.e., have been attained with the same number of reversals), the planner selects one attained by a path that minimizes the length of the curve drawn by the point P_1 of the tractor (midpoint between the two rear wheels). This choice essentially corresponds to favoring straight paths over curvy ones. The size of the search space is slightly increased, but the appearance of the output paths is much nicer.

We first experimented with the planner using a simulated car-like robot (one-body mobile robot) with several workspace arrangements:

Figure 4 shows an example of the classical parallel parking problem with a very limited steering angle ($\varphi \in [-\varphi_{\max}, +\varphi_{\max}]$, with $\varphi_{\max} = 30$ degrees). The running time for that example was 20 seconds ($R = 8$).
Figure 5 shows an example in a cluttered workspace when the maximal steering angle $\varphi_{\max}$ is 45 degrees. The running time was about 30 seconds ($R = 8$). Four reversals were necessary in this example.
Figure 6 shows a path among randomly generated obstacles for a car whose maximal steering angle is 45 degrees. The running time was about 2 minutes ($R = 9$). The generated path contains four reversals. (More precisely, the workspace bitmap for this example was obtained by setting every cell to 0 or 1 with equal probability 0.5, applying a Gaussian filter to the resulting map, and thresholding the result of the filtering operation.)
Figure 7 shows a path for a car that can only turn to the left. The two extremal values of the steering angle are $\varphi_{\min} = 22.5$ degrees (on the left) and $\varphi_{\max} = 45$ degrees (on the left). Seventeen reversals were necessary in this example. The running time was about 20 seconds ($R = 8$).

One of the authors implemented another planner for car-like robots [27]. This other planner is based on the approach proposed in [44] and [23] with substantial internal differences. In general, for simple problems (intuitively, those for which there exists a holonomic path that lies far away from the obstacles), the planner described in [27] is faster than the planner described above; it also tends to produce shorter paths, but with more reversals. For trickier problems (intuitively, those for which nonholonomic paths have to be quite different from holonomic paths, e.g., as in parallel parking), the planner described above tends to be faster and to produce paths with many less reversals (hence, much faster to execute with a real robot).

We also conducted several experiments with a tractor–trailer (two-body mobile robot):

Figure 8 shows the parallel-parking example for a tractor–trailer with a very limited steering angle ($\varphi_{\max} = 30$ degrees). The running time was 2 minutes ($R = 7$).
Figure 9 shows a path in a workspace with several obstacles. The maximal steering angle is 45 degrees. The running time was about 5 minutes ($R = 7$).
Figure 10 shows an example where the tractor–trailer has to maneuver in a cluttered workspace with a maximal steering angle equal to 45 degrees. The running time was about 10 minutes ($R = 7$).

The easiness with which the above examples have been generated empirically demonstrates the robustness of the implemented planner.

7. Conclusion. In this paper we have used results in nonlinear control theory to establish new results concerning the controllability of robots subject to constraints on the velocity. The constraints may be linear or nonlinear in the velocity. They may be equality or inequality constraints. These results have then been applied to multibody mobile robots (tractor towing a sequence of trailers rolling on a

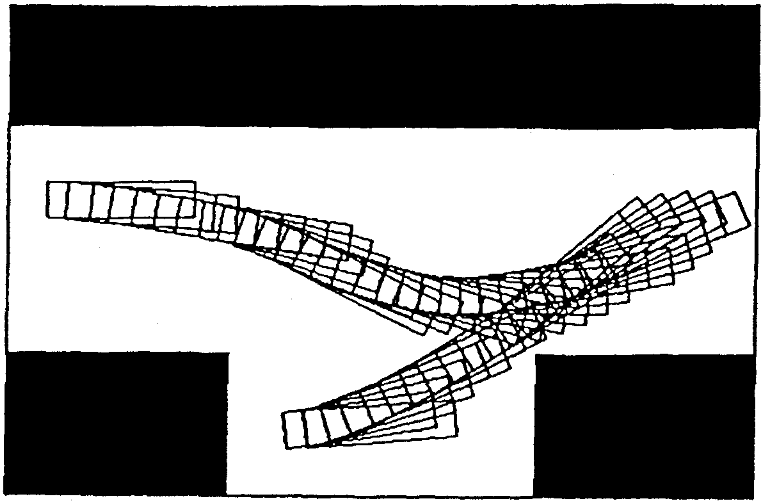

Fig. 4. Parking a car.

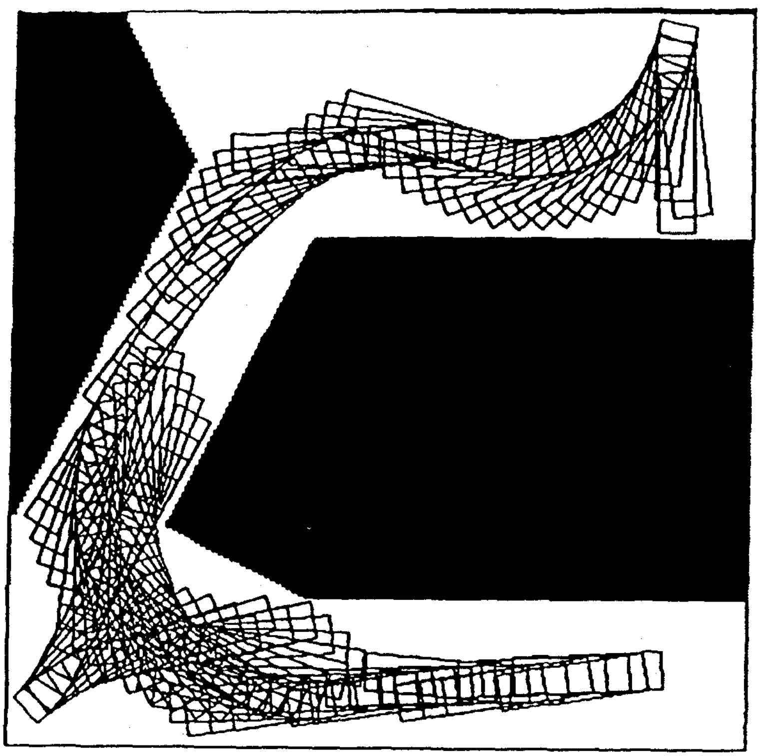

Fig. 5. Car maneuvering in a cluttered workspace.

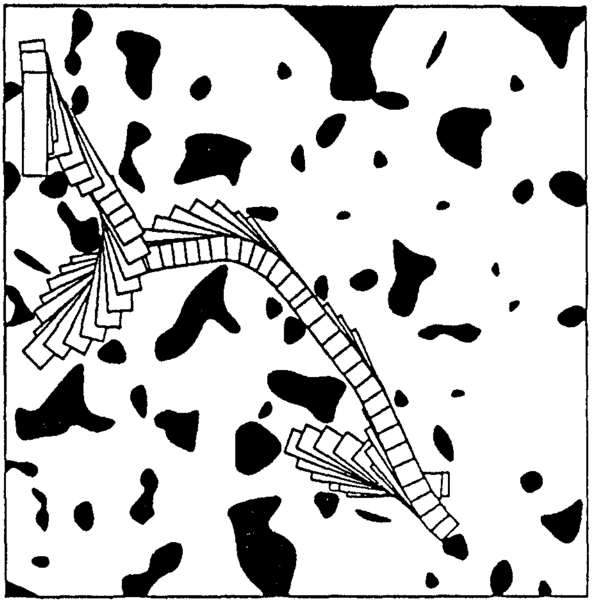

Fig. 6. Car maneuvering among randomly generated obstacles.

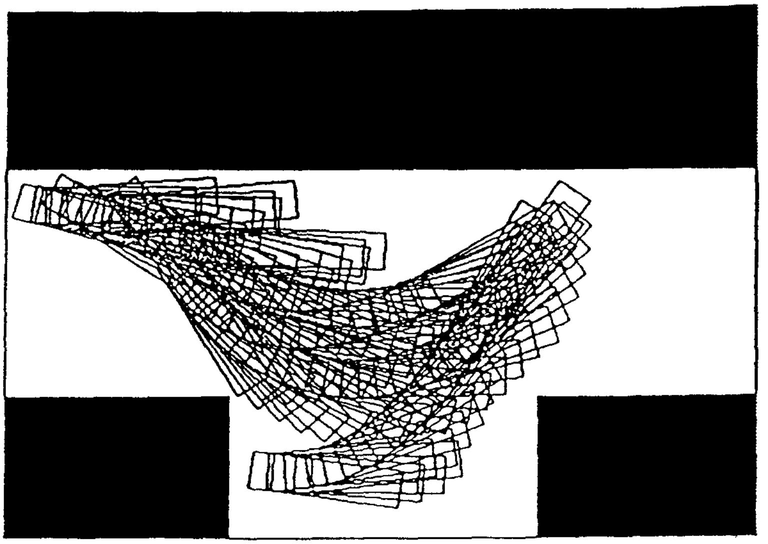

Fig. 7. Parallel parking by a car that can only turn to the left.

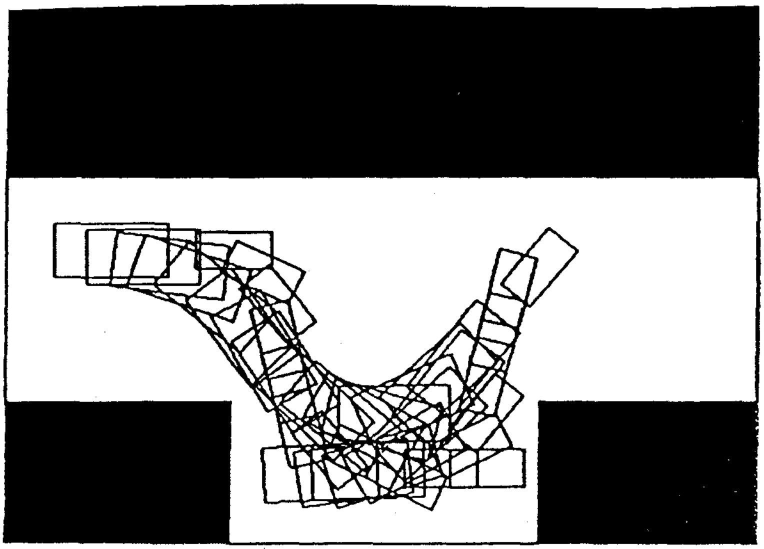

Fig. 8. Parking a tractor–trailer.

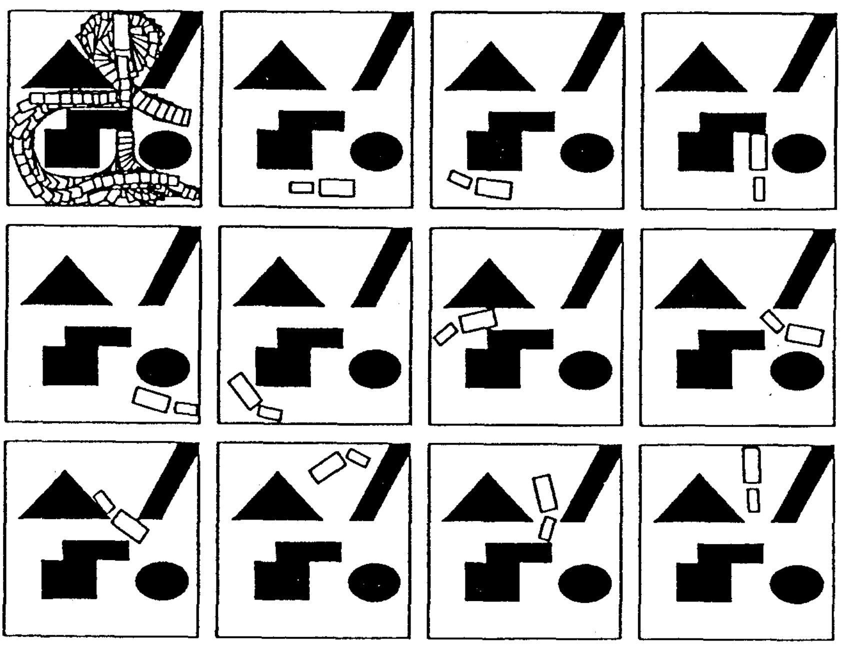

Fig. 9. Tractor–trailer maneuvering among various obstacles.

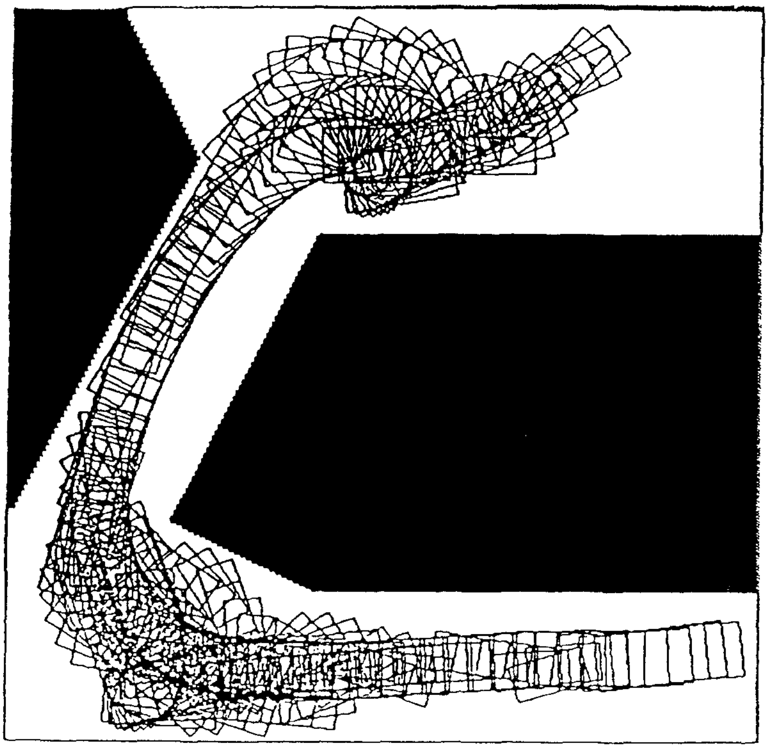

Fig. 10. Tractor–trailer maneuvering in a cluttered workspace.

plane), showing that multibody mobile robots are controllable whenever there are at least two different admissible positions of the tractor's steering wheel. We also derived the differential equations of motion for any multibody car system.

We have designed a path planner deriving from these results. The planner generates a path by exploring a finite subset of the configuration space. This subset is defined by discretizing the control parameters of the robot according to the controllability result previously obtained. In theory, this planner is general and can generate paths for mobile robots with arbitrarily many bodies. We have shown that it is asymptotically complete, i.e., for a given problem, if the discretization of the search space is set fine enough, the planner is guaranteed to find a solution path, if the problem admits a solution path. We also have shown that it is asymptotically optimal for car-like robots and, in a slightly weaker sense, for multibody mobile robots. The planner is approximate in the sense that it produces a path whose final configuration is only contained in a neighborhood of the goal configuration. This neighborhood can be defined as small as we wish as part of the planning problem.

In its current version, the planner focuses on generating paths with minimized number of reversals. In relatively rare circumstances, this may yield excessively long paths. Perhaps a good variant would be to replace the Dijkstra algorithm by a more general A^* algorithm [40] with a cost function blending the number of reversals and the length of the path and an admissible heuristic function equal to the distance to the goal. However, the optimality claims (Claims 2 and 3) would no longer be valid, even with respect to the new criterion combining the number of reversals and the length of the path.

The time/space complexity of the planner is exponential in the number of bodies of the mobile robot. For this reason, the current implementation of the planner is only practical for mobile robots with one and two bodies. Experiments with these two sorts of robots have shown that, despite the brute-force method that it uses, the planner can solve tricky problems in a reasonable amount of time. The planner could possibly be extended to higher-dimensional configuration spaces by using potential field techniques to guide the search, as described in [4] and [5]. However, so far, we have not investigated this extension. Planning efficient motions for mobile robots with more than two bodies is still an open computational problem.

Appendix A. This appendix establishes a result used in the proofs of Claims 1 and 2. In these proofs we consider a path τ connecting two configurations q_1 and q_2 and obtained by applying a finite sequence of controls

$$u_1, \dots, u_r \in \{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\},$$

with respective finite durations $\delta t_1, \dots, \delta t_r$. We show below that τ can be approximated as close as we wish by a path τ_ε obtained by applying a finite sequence of controls in $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$, each with the *same* duration δt_0. In addition, τ_ε can be constructed so that it has the same number of reversals as τ.

Without loss of generality, we assume that any two consecutive controls in the sequence of controls producing τ are different. We regard τ as the concatenation of r subpaths τ_i each executed with a single control u_i, i.e., a constant linear velocity and a constant steering angle.

Consider any subpath τ_i. Let $\tau_i(0)$ and $\tau_i(1)$ be its initial and final configurations, respectively. Starting at $\tau_i(0)$, let us apply a control sequence $u_i^{(1)}, \dots, u_i^{(k_i)}$, where $u_i^{(j)} = u_i$ for any $j \in [1, k_i]$, with each control $u_i^{(j)}$ applied with the same duration δt_0. The produced path, call it τ'_i, first coincides with τ_i, and only differs at the end (it may end slightly before or slightly after τ_i). By choosing δt_0 and k_i appropriate, we can make the endpoint of τ'_i as close as we wish to $\tau_i(1)$.

Let us now construct τ_ε as a path starting at q_1 and obtained by applying a control sequence of the form

$$u_1^{(1)}, \dots, u_1^{(k_1)}, \dots, u_r^{(1)}, \dots, u_r^{(k_r)},$$

where $u_i^{(j)} = u_i$ for any $i \in [1, r]$ and any $j \in [1, k_i]$, with each control $u_i^{(j)}$ applied with the same duration δt_0. Clearly, since r is finite and only depends on the path τ (hence, fixed), we can choose δt_0 small enough so that there exist values for $k_1, \dots, k_r$ resulting in a path τ_ε that remains closer to τ than a predefined distance $\varepsilon > 0$ and whose endpoint $\tau_\varepsilon(1)$ is closer to the endpoint $\tau(1)$ of τ than ε. In addition, by construction, τ_ε has the same number of reversals as τ.

Appendix B. Consider a multibody mobile robot whose steering angle φ takes values in $[\varphi_{\min}, \varphi_{\max}] \subset (-\pi/2, +\pi/2)$. Let q_1 and q_2 be two configurations in the same connected component of $\mathscr{C}_{\text{free}}$. Proposition 4 tells us that there exists a free path between q_1 and q_2 that only uses controls in $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$. But it says nothing about the number of reversals of such a path. For a car-like robot (one body), the following proposition establishes that there exists such a path with minimal number of reversals, over all possible free paths between q_1 and q_2. This proposition is used in the proof of Claim 2.

PROPOSITION 5. *Between any two configurations in the same connected component of the free space of a car-like robot there exists a free path with minimal number of reversals (over all possible free paths between the same two configurations) that only uses controls in* $\{-1, +1\} \times \{\varphi_{\min}, \varphi_{\max}\}$.

PROOF. Let us consider a free path τ joining the two configurations with minimal number λ of reversals. Such a path necessarily exists. When the robot moves along τ, the midpoint P_1 between the two rear wheels of the car (see Section 5) traces a curve γ whose curvature is upper-bounded [26]. At every point of γ the current orientation of the car is defined by the tangent to γ at this point. We partition γ into maximal segments γ_i, $i = 1, 2, \ldots,$ of nonzero length, each with a curvature of constant sign (negative, positive, null). The number of such segments is necessarily finite.

Let us first assume that $\varphi_{\min} < 0$ and $\varphi_{\max} > 0$. During a motion with $\varphi = \varphi_{\min}$ (resp. $\varphi_{\max}$) P_1 moves along a circle of radius $\rho_{\min} = L_1/\tan\varphi_{\min}$ (resp. $\rho_{\max}) = L_1/\tan\varphi_{\max}$). Consider any segment γ_i performed with linear velocity $+1$ with the robot turning to the right (i.e., $0 < \varphi \leq \varphi_{\max}$). We define the side of γ_i on which lies the centers of curvature to be the right side of γ_i, and the other side to be its left side. We divide γ_i into small segments $\delta_j\gamma_i$ with endpoints $p_{i,j}$ and $p_{i,j+1}$ (see Figure 11(a)). For every point $p_{i,j}$, we draw the circle $C_{i,j}$ of radius $\rho_{\max}$ that is tangent to γ_i at $p_{i,j}$ and lies on the right side of γ_i. Every $C_{i,j}$ intersects γ_i at a single point ($p_{i,j}$). We choose the points $p_{i,j}$ close enough to each other that every two successive circles $C_{i,j}$ and $C_{i,j+1}$ intersect at two distinct points (as in Figure 11(a)). Now we draw the circle $D_{i,j}$ of radius $\rho_{\min}$ that is tangent to both $C_{i,j}$ and $C_{i,j+1}$ with its center located on the left side of γ_i (or as close as possible to the left side if $\rho_{\min}$ is too small). We define $\Delta_j\gamma_i$ as the curve segment obtained by concatenating the following three circular arcs:

(1) the short circular arc in $C_{i,j}$ connecting $p_{i,j}$ to the tangent point of $C_{i,j}$ and $D_{i,j}$;
(2) the short circular arc in $D_{i,j}$ connecting the tangent point of $C_{i,j}$ and $D_{i,j}$ to the tangent point of $D_{i,j}$ and $C_{i,j+1}$; and
(3) the short circular arc in $C_{i,j+1}$ connecting the tangent point of $D_{i,j}$ and $C_{i,j+1}$ to $p_{i,j}$.

By dividing γ_i into small enough segments, every $\Delta_j\gamma_i$ can be made as close as we wish to $\delta_j\gamma_i$. As mentioned above, at every point along each of the two curve segments $\delta_j\gamma_i$ and $\Delta_j\gamma_i$, the orientation of the car is defined by the tangent to the

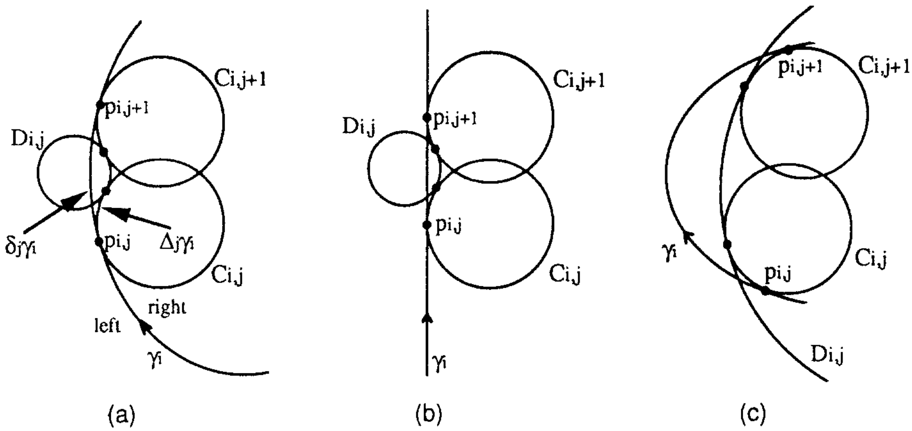

Fig. 11. Approximation of a path.

curve segment at the considered point. Therefore, since $\delta_j\gamma_i$ and $\Delta_j\gamma_i$ are tangent to each other at their two extremities, the car has the same orientation both at their initial extremities and at their final extremities. By making the $\delta_j\gamma_i$ small enough, we can make the change of orientation of the tangent along both $\delta_j\gamma_i$ and $\Delta_j\gamma_i$ arbitrarily small. Hence, since the subpath of τ that projects onto $\delta_j\gamma_i$ is contained in the open free space, by making every $\delta_j\gamma_i$ small enough, we can lift every curve segment $\Delta_j\gamma_i$ into a path segment contained in $\mathscr{C}_{\text{free}}$ and having its extremities in the original path τ. Similar constructs as above can be made for a straight segment (Figure 11(b)) or a segment that turns to the left. Standard arguments about the compactness of the path τ (the fact that any open cover of the path admits a finite subcover) allow us to derive that τ can be approximated by a finite sequence of such path segments in $\mathscr{C}_{\text{free}}$. This approximation includes the same number of reversals as τ, hence is optimal in that respect.

Similar constructs as above can be made if $\varphi_{\min}$ and $\varphi_{\max}$ have the same sign (see Figure 11(c)). □

We may wish to generalize this proposition to mobile robots with one or more trailers. The above proof does not directly work, because when the point P_1 traces a curve segment $\Delta_j\gamma_i$, the final orientation of each trailer is, in general, different from the orientation it would have had if P_1 had traced $\delta_j\gamma_i$. Of course, we could make the difference in orientation arbitrarily small by dividing γ_i into small enough segments, but the number of segments would then increase, so that we could not conclude anything.

References

[1] Aho, A. V., Hopcroft, J. E., and Ullman, J. D. *Data Structures and Algorithms*. Addison-Wesley, Reading, MA, 1983.

[2] Alexander, H. L. Experiments in Control of Satellite Manipulators. Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, 1987.

[3] Arnold, V. I. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, New York, 1978.

[4] Barraquand, J. and Latombe, J. C. Robot Motion Planning: A Distributed Representation Approach. Report STAN-CS-89-1257, Computer Science Department, Stanford University, Stanford, CA, May 1989.

[5] Barraquand, J. and Latombe, J. C. Robot Motion Planning: A Distributed Representation Approach. *The International Journal of Robotics Research,* **10**(6) (1991).

[6] Barraquand, J. and Latombe, J. C. On Nonholonomic Mobile Robots and Optimal Maneuvering. *Proc. Fourth IEEE International Symposium on Intelligent Control,* Albany, New York (1989), pp. 340–347.

[7] Barraquand, J. and Latombe, J. C. On Nonholonomic Mobile Robots and Optimal Maneuvering. *Revue d'Intelligence Artificielle,* **3**(2) (1989), 77–103.

[8] Barraquand, J. and Latombe, J. C. Controllability of Mobile Robots with Kinematic Constraints. Report No. STAN-CS-90-1317, Department of Computer Science, Stanford University, June 1990.

[9] Brooks, R. A. and Lozano-Pérez, T. A Subdivision Algorithm in Configuration Space for Find-Path with Rotation. *Proc. Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, FRG (1983), pp. 799–806.

[10] Cai, C. Instantaneous Robot Motion with Contact Between Surfaces, Report No. STAN-CS-88-1191, Department of Computer Science, Stanford University, 1988.

[11] Chow, W. L. Uber Systeme von Linearen Partiellen Differentialgleichungen ester Ordnung. *Mathematische Annalen,* **117** (1939), 98–105.

[12] Cutkosky, M. *Grasping and Fine Manipulation,* Kluwer Academic, Boston, 1985.

[13] Faverjon, B. Object Level Programming Using an Octree in the Configuration Space of a Manipulator. *Proc. IEEE International Conference on Robotics and Automation,* San Francisco (1986), pp. 1406–1412.

[14] Faverjon, B. and Tournassoud, P. A Local Based Approach for Path Planning of Manipulators with a High Number of Degrees of Freedom. *Proc. IEEE International Conference on Automation and Robotics,* Raleigh, NC (1987), pp. 1152–1159.

[15] Fortune, S. and Wilfong, G. Planning Constrained Motion. *Proc. STOCS,* ACM, Chicago (1988), pp. 445–459.

[16] Gouzènes, L. Strategies for Solving Collision-Free Trajectories Problems for Mobile and Manipulator Robots. *The International Journal of Robotics Research,* **3**(4) (1984), 51–65.

[17] Greenwood, D. T. *Principles of Dynamics,* Prentice-Hall, Englewood Cliffs, NJ, 1965.

[18] Haynes, G. W. and Hermes, H. Non-linear Controllability via Lie Theory. *SIAM Journal of Control,* **8** (1970), 450–460.

[19] Hermann, R. On the Accessibility Problem in Control Theory. *International Symposium on Nonlinear Differential Equations and Nonlinear Mechanics* (1963), pp. 325–332. Academic Press, New York.

[20] Hermann, R. and Krener, A. J. Nonlinear Controllability and Observability. *IEEE Transactions on Automatic Control,* **22**(5) (1977), 728–740.

[21] Isidori, A. *Nonlinear Control Systems: An Introduction,* Springer-Verlag, New York, 1985.

[22] Jacobs, P. and Canny, J. Planning Smooth Paths for Mobile Robots. *Proc. IEEE International Conference on Robotics and Automation,* Scottsdale, AZ (1989), pp. 2–7.

[23] Jacobs, P., Laumond, J. P., and Taix, M. Efficient Motion Planners for Nonholonomic Mobile Robots. *Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems,* Osaka, Japan (1991), pp. 1229–1235.

[24] Krener, A. J. A Generalization of Chow's Theorem and the Bang–Bang Theorem to Nonlinear Control Problems. *SIAM Journal of Control,* **12** (1974), 43–52.

[25] Lafferriere, G. and Sussman, H. J. Motion Planning for Controllable Systems without Drift: A Preliminary Report. Report SYCON-90-04, Rutgers Center for Systems and Control, Rutgers University, New Brunswick, NJ, June 1990.

[26] Latombe, J. C. *Robot Motion Planning,* Kluwer Academic, Boston, 1991.

[27] Latombe, J. C. A Fast Path Planner for a Car-Like Indoor Mobile Robot. *Ninth National Conference on Artificial Intelligence,* Anaheim, CA (1991), pp. 659–665.

[28] Laugier, C. and Germain, F. An Adaptative Collision-Free Trajectory Planner. *International Conference on Advanced Robotics,* Tokyo, Japan (1985).

[29] Laumond, J. P. Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints. *Proc. International Conference on Intelligent Autonomous Systems*, Amsterdam (1986), pp. 346–354.

[30] Laumond, J. P. Finding Collision-Free Smooth Trajectories for a Nonholonomic Mobile Robot. *Proc. Tenth International Joint Conference on Artificial Intelligence*, Milano, Italy (1987), pp. 1120–1123.

[31] Laumond, J. P., Siméon, T., Chatila, R., and Giralt, G. Trajectory Planning and Motion Control of Mobile Robots. *Proc. IUTAM/IFAC Symposium*, Zurich, Switzerland (1988).

[32] Laumond, J. P. and Siméon, T. Motion Planning for a Two Degrees of Freedom Mobile Robot with Towing. Technical Report No. 89-148, LAAS/CNRS, Toulouse, 1989.

[33] Laumond, J. P. Nonholonomic Motion Planning Versus Controllability Via the Multibody Car System Example. Report STAN-CS-90-1345, Department of Computer Science, Stanford University, December 1990.

[34] Li, Z. and Canny, J. F. Robot Motion Planning with Nonholonomic Constraints. Memo UCB/ERL M89/13, Electronics Research Laboratory, University of California, Berkeley, February 1989.

[35] Li, Z., Canny, J. F., and Sastry, S. S. On Motion Planning for Dexterous Manipulation, Part I: The Problem Formulation. *IEEE International Conference of Robotics and Automation*, Scottsdale, AZ (1989), pp. 775–780.

[36] Lobry, C. Contrôlabilité des systèmes non lineaires. *SIAM Journal of Control*, **8** (1970), 573–605.

[37] Lozano-Pérez, T. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, **32**(2) (1983), 108–120.

[38] Lozano-Pérez, T. A Simple Motion-Planning Algorithm for General Robot Manipulators. *IEEE Journal of Robotics and Automation*, **3**(3) (1987), 224–238.

[39] Nakamura, Y. and Mukherjee, R. Nonholonomic Path Planning and Automation, Scottsdale, AZ (1989), pp. 1050–1055.

[40] Nilsson, N. J. *Principles of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA, 1980.

[41] Poston, T. and Stewart, I. *Catastrophe Theory and its Applications*, Pitman, London, 1978.

[42] Reeds, J. A. and Sheep, R. A. Optimal Paths for a Car that Goes both Forward and Backward. *Pacific Journal of Mathematics*, **145**(2) (1991), 367–393.

[43] Sussmann, H. J. and Jurdjevic, V. J. Controllability of Nonlinear Systems. *Journal of Differential Equations*, **12** (1972), 95–116.

[44] Taix, M. Planification de Mouvement pour Robot Mobile Non-Holonome. Doctoral Dissertation, LAAS, Toulouse, France, January 1991.

[45] Tournassoud, P. and Jehl, O. Motion Planning for a Mobile Robot with a Kinematic Constraint. *Proc. IEEE International Conference on Robotics and Automation*, Philadelphia (1988), pp. 1785–1790.

[46] Warner, F. W. *Foundations of Differentiable Manifolds and Lie Groups*. Springer-Verlag, New York, 1983.

[47] Wilfong, G. Motion Planning for an Autonomous Vehicle. *Proc. IEEE International Conference on Robotics and Automation*, Philadelphia (1988), pp. 529–533.