Nonholonomic Motion Planning versus Controllability via the Multibody Car System Example

by

Jean-Paul Laumond

Department of Computer Science

Stanford University
Stanford, California 94305

Nonholonomic Motion Planning versus Controllability via the Multibody Car System Example

Jean-Paul Laumond *

Robotics Laboratory
Department of Computer Science
Stanford University, CA 94305

(Working paper)

Abstract

A **multibody car system is a** *non-nilpotent, non-regular, triangularizable* **and** *well-controllable* **system. One goal of the current paper is to prove this obscure assertion.** But its main goal is to **explain and enlighten what it means.**

Motion planning is an already old and classical problem in Robotics. A few years ago a new instance of this problem has appeared in the literature : *motion planning for nonholonomic systems.* While useful tools in motion planning come from Computer Science and Mathematics (Computational Geometry, Real Algebraic Geometry), nonholonomic motion planning needs some Control Theory and more Mathematics (Differential Geometry).

First of all, this paper tries to give a computational reading of the tools from Differential Geometric Control Theory required by planning. Then it shows that the presence of obstacles in the real world of a real robot challenges Mathematics with some difficult questions which are topological in nature, and have been solved only recently, within the framework of Sub-Riemannian Geometry.

This presentation is based upon a reading of works recently developed by Murray and Sastry [39], Lafferiere and Sussmann [55], and Bellaiche, Jacobs and Laumond [5] [33].

Contents

1 Introduction

The motion planning problem (MP) is certainly one of the best formulated ones in Robotics. It raises two questions :

- Can a robot reach a given goal while avoiding the obstacles of its environment ? This is the *decision* problem.

- If the answer to the previous question is yes, ***what path*** may it follow ? This is the ***complete*** problem.

The geometric formulation of this problem is known as the piano mover problem. This formulation considers the motion of rigid bodies amidst obstacles in the 3-dimensional Euclidean space. The placement (translation and rotation) of a body in the Euclidean space is given by a point in a 6-dimensional space. Some geometric relationship between the bodies may appear for a given robotic system (as is typical for a robot arm). They are translated into equations between the placement parameters of the bodies. These are called the ***holonomic links.*** They restrict the space of the allowed placements to a subspace of the placement spaces of all the bodies. This subspace is called the ***configuration space.*** Finally, a configuration of the robot is represented by a point of the configuration space that defines precisely the domain occupied by the robot in Euclidean space. A point-path in the configuration space corresponds to a motion of the robot. For a holonomic system, we have as many degrees of freedom as is needed to follow ***any*** path.

Therefore, for holonomic systems, the existence of a collision-free trajectory is characterized by the existence of a connected component in the admissible (i.e., collision-free) configuration space. To solve MP, it is enough to compute the admissible configuration space (i.e., transform the obstacles in the Euclidean space into "obstacles" in the configuration space), and then explore its connected components.

Since the seventies this problem has attracted many researchers working in Robotics and beyond, in Computational Geometry and Algebraic Geometry. See [48] for a recent overview, [60] for a review of the various approaches of the problem, and [27] as the first genuine book on the subject .

However, there are cases in which this formulation of motion planning is not sufficient. In the last four years, an example of the limitation of the piano mover formulation has been investigated : planning constrained motions where constraints are nonholonomic in nature. A ***nonholonomic link*** is expressed as a non-integrable equation involving derivatives of the configuration parameters. Such constraints are expressed in the tangent space at each configuration; they define the allowable velocities of the system, and they ***cannot*** be eliminated by defining a more restricted configuration space manifold. Thus, the main consequence of a nonholonomic constraint is that an arbitrary path in the admissible configuration space does not necessarily correspond to a feasible trajectory for the robot. Therefore, the existence of a collision-free trajectory is not ***a priori*** characterized by the existence of a connected component in the admissible configuration space.

Planning motions for nonholonomic systems is not as new in other communities as in the community working on obstacle avoidance for robots[1]. This problem is well known in Nonlinear

---

[1] **Notice that nonholonomic motion planning appears also in some spatial applications, for systems (like space-stations or satellite) using internal motion and submitted to conservation laws (see [40] [42]).**

Control and in Differential Geometry. Important results have been obtained over the last two decades, while the first results seem dated from the thirties with Chow's work [12].

Notwithstanding, Robotics brings to the front an important constraint which is not usually taken into consideration : the planner has to produce trajectories that *avoid the obstacles.* Moreover, by its applications to the real world, it requires effective and efficient computational tools, rather than just proofs of existence.

Results useful to our problem can be found in publications-often difficult ones-from other communities than the Robotics one. Because the viewpoints are different, they at tack only some aspects of the problem, and use different terminologies. The goal of this paper is to enlighten these points of view by a computational one (the right point of view for the planning problem) and to stress the connections between motion planning and differential geometric control theory. This study has to be viewed as an informal statement of these connections, hopefully readable by a non-specialist in control theory or in differential geometry (as the author is). A spotlight will be focused on some concepts from these theories, using a minimal formalism while trying to understand where and why these are pertinent as far as the motion planning problem is concerned. It is clear that an in-depth study of these concepts needs the precision given by the mathematical formalism : for each concept there will be references introducing or using it.

Along this study we take a multibody car system (i.e., like a luggage carrier in an airport) as an example of application. The results (Section 3 and Section 5) for this example constitute a contribution by themselves. They can be read independently of the rest.

The decision problem of planning for nonholonomic systems is related to their controllability[2] : more precisely the existence of a collision-free trajectory for a controllable nonholonomic robot is characterized by the existence of an *open* connected component of the admissible configuration space. The decision problem is then similar to that of the piano mover problem. This result constitutes the first main link; it has been studied simultaneously in several research groups : [31] [34] [3] [38]. It is based upon the Lie algebra rank condition, and will be recalled in Section 2.

Section 3 presents a three-step modeling of the multibody car system, from a purely geometric model leading to the definition of the configuration space (Section 3.1), to four distinct control models, all corresponding to practical applications (Section 3.3), via a differential model (Section 3.2). The differential model finally enables us to give a unified proof of controllability encompassing all the envisaged systems (Section 5).

Section 4 refines the presentation of Section 2, by considering the computational aspects of the problem : is a system controllable ? There is a semi-decidable procedure for this problem, which is supported by several concepts of differential geometric control theory. Then this paper introduces the *well-controllability* notion, in relation to the planning problem.

Nevertheless, at this stage the complete problem of nonholonomic motion planning remains unsolved.

---

[2]The use of the term "controllability" in this context is fuzzy in the community. Indeed, the meaning we use here is related to the reachability concept. A nonholonomic system may be controllable by *open* loops. It does not have to be controllable by *closed* loops (see [47] for a study of feedback controls for a nonholonomic wheeled cart). It would be better to use the notion of *a completely nonholonomic system* related to the concept of a distribution (see [58]). This paper adheres to Sussmann's terminology [54], which seems to have reached some state of general acceptance

Section 6 considers the complete problem. We will see that the key question is *topological* in nature. While motion planners for nonholonomic systems have blossomed through the last two years (see Appendix 1), very recent contributions pursue a deep study of the differential geometric tools available for solving the complete problem. [33] presents an efficient planner for mobile robots and shows that its strategy can be generalized. [55] presents a general planner based upon a general constructive proof of controllability. These results let us stress the main difficulty for building efficient planners : while obstacle avoidance requires us to consider the "natural" Riemannian topology of the configuration space (i.e., induced by the natural Hausdorff metric working in the robot environment), the trajectories allowed by the nonholonomic constraints compel us to consider another topology in this space : the topology induced by the length of the shortest *allowed* path between two points. Such a metric is known as a sub-Riemannian (or singular, or Carnot-Caratheodory) metric. In fact, using sub-Riemannian geometry (see [7] [51] [58] [36]), it is possible to show that both topologies are the same. This result enables us to conclude on the generality of the approaches presented in [33] and [55]. Nevertheless, because we are interested in the computational point of view, we have to study more deeply the shape of the sub-Riemannian metrics in order to estimate the combinatorial complexity of the planners. This study has been done in [33] for the case of the car-like robot. This section points at the reference [58] that gives precisely the general and finest form of the sub-Riemannian metric we need to conclude on the complexity of the complete problem.

Appendix 1 overviews other results related to nonholonomic motion planning : they do not use intensively the tools we present in this paper, but they are interesting nonetheless from either a theoretical or a practical point of view. Following appendices give the various computations for the examples presented along the paper (Appendix 2), together with the tedious proof of the controllability result presented in Section 5 (Appendix 3).

Two starting points are at the origin of this working paper. The first one has been the work of Barraquand and Latombe on the controllability of multibody systems (see [4]). This report yields a proof of controllability in a general case involving up to n trailers. The second point has been a reading of papers written by Murray and Sastry [38] [39] and by Lafferiere and Sussmann [55]. It seemed interesting to clarify the relationship between these approaches of the planning problem, taking also into account the approach ([30] [33]) developed within the Hilare mobile robot project [16] [10] by Jacobs and the author.

2 From Planning to Control

Until a very recent period, the main contribution to nonholonomic motion planning (independently developed in [31] and [34]) has been to solve the decision part of the nonholonomic motion planning problem, via differential geometry and control theory. See [3] for a clear presentation of the necessary tools that we examine here.

While the constraints due to the obstacles are expressed directly in the manifold of configurations, nonholonomic constraints deal with the tangent space. In the presence of a link between the robot's parameters and their derivatives, two questions arise :

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

- If not, does it reduce the accessible configuration space ?

In the case of r links corresponding to r equations linear in the derivatives of the n parameters, these equations determine what is called an $(n-$ r)-distribution A on the manifold of configurations. The answer to the first question is then given by Frobenius' theorem (see for instance [49]): the equations are integrable if and only if the distribution A is closed under the Lie bracket operation. Let us recall that the Lie bracket of two vector fields X and Y is defined as $[X, Y] = \partial X.Y - \partial Y.X$. A sample of computation examples appears in Appendix 2.

From a control theory perspective, a control is a function which allows us to choose the system state velocity at each instant by a careful weighting of smooth vector fields. The control Lie algebra associated with A, denoted by *LA(A),* is the smallest distribution which contains A and is closed under the Lie bracket operation. The answer to the second question is then given by the non-linear system controllability theorem (see for instance [53] [35] [17]): if the rank of the Lie algebra is full at a given configuration c, then there exists a neighborhood $\mathcal{N}$ of c whose points represent configurations reachable by the system moving from c along an admissible path. Moreover, this path stays in $\mathcal{N}$. This condition is known as the "rank condition"; it is a *local* condition.

If the rank condition holds everywhere in the configuration space, then the robot is termed *controllable*[3]. From our planning point of view, the main consequence is that the existence of a collision-free trajectory is characterized by the existence of a connected component in the *free* (i.e., with neither collision nor contact) configuration space.

Therefore, apart from topological subtleties dealing with motions in contact, **the decision problem of motion planning for controllable systems is the same as for holonomic ones**[4].

The difference is more involved with the complete problem. The previous result answers the question of the existence of a feasible trajectory, that is, the decision problem, but does not solve the

---

[3]In fact a controllable system requires only that the rank condition holds nearly everywhere, that is on a dense subset of the manifold. Various precise controllability concepts appear involving the "size" of reachable sets and the "size" of sets where the rank condition holds (see [54]). Omitting them does not affect our purpose.

'Section 3 highlights the difficulty for proving the controllability of a system. Since we are interested in the computational point of view, we are looking for a procedure that allows us to conclude. As we will see, this problem is not trivial.

problem of efficiently producing an admissible trajectory. To the extent of the author's knowledge, there was no general **constructive** proof of the result mentionned above until the recent contribution of Lafferiere and Sussmann [55][5]. Their proof leads to the design of a general nonholonomic motion planner in the absence of obstacles (see Section 6).

At this stage we can just hope that the search for a solution for a nonholonomic system can be guided by a collision-free trajectory for the associated holonomic system. Indeed, thanks to the local property above, a controllable robot can be steered close to any path as long as there is a "small gap" between the reference path and the obstacles. This idea is precisely the basis of the two strategies defined in [55] and [33] that we will study in Section 6. It appears clearly that the key question for developing such a strategy deals with the size of the "gap", i.e., with the **topology** induced by the nonholonomic constraints. This aspect is detailed in Section 6.

Now let us define the multibody car system.

---

[5]Specific constructive proofs appear in [28] for the car-like robot and in [32] for the car-like robot pulling a trailer.

3 The Modeling of a Multibody Car System

Figure 1 shows what we call a multibody car system. It corresponds to a car-like robot pulling and pushing trailers (like a luggage carrier in an airport). In order to grasp the planning point of view, we will build along three modeling levels for this system : a geometric model (Section 3.1), a differential model (Section 3.2) and a control model (Section 3.3) respectively. Then a close examination shows that four different control systems correspond to the same differential model; this is the differential model that we will use to solve the decision part of the planning problem (Section 5).
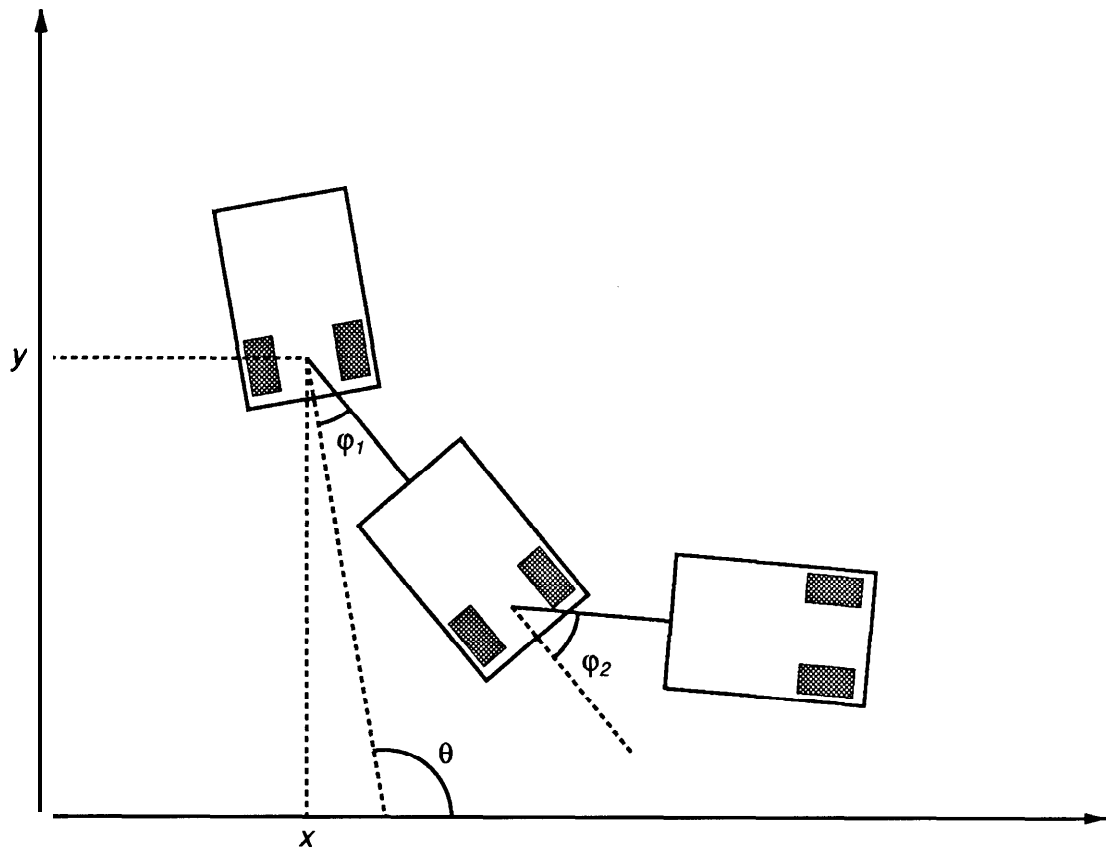

Figure 1: A multibody car system

3.1 Geometric Model

Let us consider a multibody system (Be, $\mathcal{B}_1, \ldots, \mathcal{B}_n$) constituted by a car $\mathcal{B}_0$ and n trailers $\mathcal{B}_i$. We take the midpoint between the rear wheels as the reference point for each body; its coordinates are

denoted by x_i and y_i in some fixed Cartesian frame of the plane; the angle θ_i is taken between the main axis of $\mathcal{B}_i$ and the x-axis of the frame. The space of possible placements of the $n + 1$ bodies is then $3(n + 1)$-dimensional.

In order to form a convoy (e.g., the car pulling the n trailers $\mathcal{B}_i$, each hooked up to the next like for a luggage carrier), each trailer $\mathcal{B}_i$ is assumed to be hooked up to the midpoint between the rear wheels of the preceding body $\mathcal{B}_{i-1}$[6]. This means that the system is submitted to $2n$ holonomic links which yield the following equations in the placement space[7] :

$$x_i - x_{i-1} = -\cos\theta_i,$$
$$y_i - y_{i-1} = -\sin\theta_i.$$

The configuration space of the convoy is a submanifold of dimension $3(n + 1) - 2n = n + 3$ in the placement space of all the bodies. A possible parameterization of this submanifold is to pick out the $(n + 3)$-tuple $(x_0, y_0, \theta_0, \theta_1, \ldots, \theta_n)$ belonging to $\mathbf{R}^2 \times (\mathbf{S}^1)^{n+1}$. For the sake of simplicity, we will set $x = x_0$, $y = y_0$, $\theta = \theta_0$ and $\varphi_i = \theta_i - \theta_{i-1}$, so that φ_i is the angle between the axes of $\mathcal{B}_i$ and $\mathcal{B}_{i-1}$, and use $(x, y, \theta, \varphi_1, \ldots, \varphi_n)$ as an alternative parameterization.

3.2 Differential Model

Each body is rolling on the ground without sliding; thus it is submitted to the classical nonholonomic link :

$$\dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i = 0.$$

Remark : This equation is obvious. We just want to mention the following fact : it has been obtained **without any reference to a control system;** we have just used some elementary kinematic considerations (i.e., a moving rigid body has only one instantaneous center of rotation).

The system is subject to $n + 1$ such constraints. Since the number of degrees of freedom of a mechanical system is defined as the difference between the dimension of its configuration space and the number of independent nonholonomic links, the number of degrees of freedom of our convoy is $(n + 3) - (n + 1) = 2$ (obviously, the 2 degrees of freedom of the car. . .).

These constraint equations are expressed in the tangent space of the placement space of all the bodies. In order to translate them into the tangent space of the configuration space, we have to get rid of the variables $\dot{x}_i$ and $\dot{y}_i$ for i $\neq$ 0' . By combining these $n + 1$ equations with the derivatives of the $2n$ holonomic equations, we obtain the following system of $n + 1$ linear equations :

$$\dot{x}_0 \sin\theta_i - \dot{y}_0 \cos\theta_i - \sum_{j=1}^{\cdot} \dot{\theta}_j \ \cos(\theta_j - \theta_i) = 0.$$

A configuration c being given, this system defines a plane in the tangent space at this point; this means that, for a possible motion passing through c, the velocity vector at c-whenever defined—has to lie in that plane[8]. If we compute the solution of the resulting system, we obtain :

---

[6] The hooking system is important. The equations do not simplify if the trailers are hooked behind the rear axles. -Moreover the problem is unsolved for this generalization.

[7] We assume that all the links have the same length 1; obviously, this does not affect the generality of the following results.

[8] The set of all these planes has the structure of what is called a distribution on the manifold.

$$\dot{x}_0 = \alpha \cos\theta_0$$
$$\dot{y}_0 = \alpha \sin\theta_0$$
$$\dot{\theta}_0 = \beta$$
$$\dot{\theta}_1 = \alpha \sin(\theta_1 - \theta_0)$$
$$\dot{\theta}_2 = \alpha \cos(\theta_1 - \theta_0)\sin(\theta_2 - \theta_1)$$

$$\cdot$$

$$\dot{\theta}_n = \alpha \sin(\theta_n - \theta_{n-1}) \prod_{i=1}^{n-1} \cos(\theta_i - \theta_{i-1})$$

with α and β any two reals. Setting $(\alpha, \beta) = (1,0)$ and $(\alpha, \beta) = (0,1)$ yields a basis of the plane, namely :

$$X_a = (\cos\theta_0,\ \sin\theta_0,\ 0,\ \sin(\theta_1 - \theta_0),\ \cos(\theta_1 - \theta_0)\sin(\theta_2 - \theta_1),\ \ldots$$

$$\ldots,\ \sin(\theta_n - \theta_{n-1}) \prod_{j=1}^{n-l} \cos(\theta_j - \theta_{j-1})\Big),$$

$$X_b = (0,\ 0,\ 1,\ 0,\ \ldots\ ,0).$$

If we use the change of variable $\varphi_i = \theta_i - \theta_{i-1}$, the same computation yields[9] :

$$\dot{x} = \alpha \cos 0$$
$$\dot{y} = \alpha \sin\theta$$
$$\dot{\theta} = \beta$$
$$\dot{\varphi}_1 = \beta - \alpha \sin\varphi_1$$
$$\dot{\varphi}_2 = \alpha\,(\sin\varphi_1 - \cos\varphi_1 \sin\varphi_2)$$
$$\dot{\varphi}_3 = \alpha \cos\varphi_1\,(\sin\varphi_2 - \cos\varphi_2 \sin\varphi_3\,\mathrm{j}$$

$$\cdot$$

$$\dot{\varphi}_n = \alpha\,(\sin\varphi_{n-1} - \cos\varphi_{n-1} \sin\varphi_n) \prod_{i=1}^{n-2} \cos\varphi_i$$

as the parameterized equations of the plane, and

$$X_a = \Big(\cos\theta,\ \sin\theta,\ 0,\ -\sin\varphi_1,\ \sin\varphi_1 - \cos\varphi_1 \sin\varphi_2,\ \ldots$$

$$\ldots,\ (\sin\varphi_{n-1} - \cos\varphi_{n-1} \sin\varphi_n) \prod_{j=1}^{n-2} \cos\varphi_j\Big),$$

$$X_b = \Big(0,\ 0,\ 1,\ 1,\ 0,\ \ldots\ ,0\Big)$$

Unfortunately the general shape appears only from the 6th coordinate

11

as a distinguished basis.

Recall that all this system modeling has been done without reference to any control system. It has been built just from the non-sliding hypothesis applied to each body of the convoy. So any control system will apply to the framework above. We will use the generality of the differential model in order to prove the controllability of the convoy for several control systems.

3.3 Control Models

At this moment, we have not mentioned yet *how* the system moves. We have just suggested that the first body drives the convoy. But we can imagine that the car is in the middle of the convoy; better it may be possible that there is no car, and just some effectors controlling the angular joints between the bodies (as in some japanese snake-robots). We examine here four examples of control systems and prove that, in all cases, the proof of controllability of these systems amounts to computing exactly the same Lie algebra, namely the Lie algebra spanned by the vector fields X_a and X_b introduced above.

3.3.1 The Convoy is Driven by Gofer

The Stanford mobile robot Gofer [1 1] is very interesting for our purpose. Indeed it leads to introducing a distinction between the control variables and the parameters which is pertinent from the point of view of the planning. Gofer's locomotion system is clever : three parallel driving wheels are linked together by some mechanical system; their velocities are controlled by a same motor (which produces forward or backward motions), while a second motor can control their directions. Such a design requires that there be a part of the vehicle whose direction remains fixed.

If the non-rotating part has a *geometric shape* whose projection on the plane contains the projection of the whole robot, then the robot appears, from the planning point of view, as a translating, but non-rotating body. In this case the reference frame of the robot is linked to the fixed part and the configuration space is only 2-dimensional.

Otherwise, if the projection of the non-rotating part is included in the projection of the other parts, the robot appears (always from the planning point of view) as a 2-dimensional translating and rotating body. The configuration space is thus 3-dimensional. The robot looks closely like what is called a *unicycle :* this means that the speed of the vehicle and its direction are directly controlled by two independent motors".

Therefore, with (x, y, 0) designating a configuration of the robot, the control system is :

---

[10]The distinction we have introduced is meaningless for the true robot, since Gofer is circular ! However, this is an example of the subtle links that may appear between a geometric model for planning and a differential model for control : think of the case where none of the projected parts would be included in the other; in that case the planning problem would concern a translating, rotating and deformable 2-dimensional body. As a matter of fact, Gofer is but a pretext to introduce the interesting case of the unicycle. . . And, in order to be exhaustive with the Stanford Robotics Laboratory, there is also a robot called Mobi whose sophisticated locomotion system endows with the privilege of being one of the few existing holonomic mobile robots.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2.$$

Thus if we attach n trailers at Gofer's rear its control system expresses directly as the two vector fields X_a and X_b defined in the preceding section, so that the controllability of the system will be established by computing the Lie algebra spanned by $\{X,, X_b\}$.

3.3.2 The Convoy is Driven by Hilare

Consider now the Hilare family, dwelling at Laas [16] [10]. The three mobile robots have the same classical locomotion system : two parallel driving wheels, the acceleration of both being controlled by two independent motors. Assume that the distance between the wheels is 2, that the reference point of the robot is the midpoint of the wheels and that the main direction of the vehicle is the direction of the wheels. With v_1 and v_2 as their respective speeds, the control system is :

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

In fact, from the planning point of view that we take in this paper (the speed at a configuration is not considered as a part of the problem), this system can be rewritten in the following way :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 1 \end{pmatrix} v_1 + \begin{pmatrix} \cos\theta \\ \sin\theta \\ -1 \end{pmatrix} v_2.$$

If we add the trailers, the vector fields corresponding to the controls are clearly $Y_a = X_a + X_b$ and $Y_b = X_a - X_b$. In order to prove the controllability of the convoy we have to compute the dimension of the Lie algebra spanned by $\{Y_a, Y_b\}$. Due to the bilinearity of the Lie bracket, it is equivalent to start with $\{X,, X_b\}$.

3.3.3 The Convoy is Driven by a Real Car

The case of a real car is a little bit more complicated. Indeed, in addition to the nonholonomic link, a specific constraint appears : the turning radius is lower bounded (we will see that this constraint does not cause any trouble).

From the driver point of view, a car has two degrees of freedom : the accelerator and the steering wheel (the brake is a kind of reverse accelerator and the clutch pedal is reserved to European

13

drivers . . .). Take the midpoint of the rear wheels as the reference point. Assuming that the distance between the rear and the front axle is 1, we denote by v the speed of the car and by φ the angle between the front wheels and the main direction of the car [11]. Simple arguments show that the control system is the following :

$$
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} v \cos \varphi \cos \theta \\ v \cos \varphi \sin \theta \\ v \sin \varphi \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2.
$$

Again we are only interested in the geometric planning problem. Then, since the position of the front wheels is not relevant to the problem, as well as the speed of the vehicle, the system simplifies into :

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

This is a non-linear system whose controls are v and φ. Anyway, as shown by the form of the equation, if we add trailers, the controllability of the convoy will still be proven by a close study of the Lie algebra spanned by X_a and X_b.

Remark : We did not take the constraint on the turning radius into account. This constraint can be expressed as $|\varphi| \le \varphi_0$, where φ_0 is a strictly positive real. From the point of view of the complete control model, this constraint has exactly the same meaning as an obstacle in the environment (an obstacle in the environment constrains the variables x, y and 0 to lie in some domain); therefore, it does not affect the controllability of the complete system and, consequently, it does not affect the controllability of the simplified system either[12].

3.3.4 The Convoy is Oddly Driven

The following system is no longer classical. Unlike the previous ones the driver system does not involve the first body alone but also the second one. These bodies being articulated, a configuration will be denoted by (x, y, 0, φ). The specificity of the system lies in its controls : the linear speed of the first body, and the angle φ between the bodies. This system actually corresponds to some vehicles used in civil engineering[13].

---

[11] More precisely, the front wheels being not exactly parallel (else they would slide), we take the average of their angles as the turn angle.

[12] Barraquand and Latombe [4] elaborate on this point and take advantage of it to exhibit a motion planner that can even manage constraints of the form $0 < \varphi_1 \le \varphi \le \varphi_2$ (i.e., the car can only turn right !); independently Bellaiche, Jacobs and Laumond give a proof in [5] for a Hilare-like system verifying $|v_1| \le |v_2|$, e.g., equivalent to a car-like robot: the vehicle turns ($|v_2| > 0$) only if its linear speed is not zero ($|v_1| > 0$).

[13] R. Hurteau from Polytechnical Scholl of Montreal mentioned this example to the author: that kind of vehicle is used in the conveyance of ore in some Canadian salt mines.

Let u_1 be the speed control of the first body; let u_2 be the control of the angular joint of the bodies. Since both are independent, u_1 is applied onto a vector field $Y_a = (\cos\theta, \sin\theta, a, 0)$ while u_2 is applied onto a vector field $Y_b = (0, 0, b, 1)$. A close study of the differential model above (section 3.2) leads us to find $a = \sin\varphi$ and $b = 1$. The control system is :

$$
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ \sin\varphi \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}_0 u_2 .
$$

Therefore the system is controllable if the Lie algebra spanned by Y_a and Y_b is four-dimensional. Furthermore, Y_b is equal to the vector field X_b instantiated with only one trailer. Moreover :

$$
Y_a = \begin{pmatrix} \cos\theta \\ \sin 0 \\ \sin\varphi \\ 0 \end{pmatrix}
$$

$$
= \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\sin\varphi \end{pmatrix} t \quad \sin\varphi \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}
$$

$$
= X_a + \sin\varphi X_b .
$$

Hence, Y_a ends up to be a linear combination of X_a and X_b. The Lie algebra spanned by Y_a and Y_b is the same as the Lie algebra spanned by X_a and X_b. These results obviously hold if we add trailers to this strange vehicle and the controllability of the convoy remains subject to the computation of the Lie algebra spanned by X_a and X_b as in the previous cases.

Conclusion

The decision part of the planning problem for the four multibody systems above is *the same.* We just have to study the distribution appearing in the differential model. This study appears in Section 5. We now discuss some computational aspects of the general tools we can use to achieve this.

4 From Control to Planning : a Computational Point of View

4.1 What is the Problem ?

Proving the controllability of an n-dimensional system using the rank condition involves showing that, for any point c in the manifold, there exists a family of **n** vectors fields in the Control Lie Algebra that spans $\mathbf{R}^n$ when applied to c. This stirs two difficulties due to the *local* and *global* characteristics of the problem :

- At any specific point c, finding such a family enables us to conclude. Not being able to find a suitable family does not imply that there is none. An exhaustive enumeration of possible families is impossible since there is an infinity of potential choices. We will see (Section 4.2) that this number can be reduced to a countable one, but not further, leading to the design of some semi-decidable procedures. We will proceed then to giving an estimate of the complexity of procedures testing the controllability of a system at a point.

- One may succeed in finding bases that work somewhere, but not necessarily everywhere. There may be some singularities. An interesting problem is to know whether such singularities have an intrinsic nature, or depend upon the choice we make (Sections 4.4 and 4.6).

As far as we know, testing controllability is not a decidable problem. Nevertheless, the procedure we define always terminates provided we don't encounter any singularities (Sections 4.7 and 4.8).

The material of this section uses the concepts of a distribution, also known as a Pfaffian system (see for instance [58]), and of the Free Lie Algebra (see [6]).

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

Consider the *(n − r)*-distribution A associated with a robotic system. We want to define an *algorithm* for testing the controllability of that system at a point. Precisely, we are interested in the rank of LA(A) (i.e., the distribution spanned by all the combinations of Lie brackets of vector fields in A). We can consider a basis $\mathcal{X}$ of A together with all the combinations of Lie brackets built upon that basis.

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

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

It verifies :

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

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

Remark : Such a construction can be viewed as a "breadth-first" construction. Some authors [58] [57] use another construction. A is denoted by $\tilde{\Delta}_1$. Then $\tilde{\Delta}_i$ is defined by :

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

Again :

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

Such a construction can be viewed as a "depth-first" construction. Using skew-symmetry and the Jacobi identity, we may verify that both constructions are the same[14]. We will prefer the first presentation that corresponds exactly to the concept of Phillip Hall families introduced below.

On the other hand, at a point c of our manifold (the configuration space), $(n - r) \leq$ rank A;(c) $\leq$ n [15]. Moreover, if $\Delta_i(c) \neq \Delta_{i-1}(c)$, then rank $\Delta_i(c) >$ rank $\Delta_{i-1}(c)$. Hence, if we consider the construction *locally* (i.e., by applying the distributions at a point), we can conclude that there exists an index p_c such that $\Delta_{p_c-1}(c) \neq$ A,,(c) $= \Delta_{p_c+1}(c) = \cdots$. The construction always stabilizes. The index p_c is the *degree of nonholonom y* of the system at c. Therefore a system is controllable at c if and only if rank $\Delta_{p_c} = n$ (if $p_c = 1$ we are locally in the holonomic situation). Notice that, from a global point of view, this stabilization property is not true, since the degree of nonholonomy may change from point to point. A close analysis of possible singularities shows that this degree may be arbitrarily high at singular points-even when we start with a regular distribution, the filtration we build may acquire some weird singularities. So, the degree of nonholonomy may be unbounded when c varies.

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

4.2 A Controllability Algorithm

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

$$\mathcal{X}_1 = \mathcal{X}$$
$$\mathcal{X}_i = \mathcal{X}_{i-1} \bigcup_{j+k=i} [\mathcal{X}_j, \mathcal{X}_k]$$

---

[14] For example, take [[X, Y], [X, [X, Z]]], an element of Δ_5:

$$[[X,Y],[X,[X,Z]]] = -[X,[X,[Y,[X,Z]]]] + [X,[Y,[X,[X,Z]]]] + [X,[Z,[X,[X,Y]]]] - [Z,[X,[X,[X,Y]]]].$$

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

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

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

$$\mathcal{X}_1 \text{ c } \mathcal{X}_2 \text{ c } \text{x3} \text{ , } \cdots \subset \mathcal{LA}(\mathcal{X})$$

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

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

4.2.1 P hillip Hall Families

In this section[16] the elements of **LA(X)** are considered as formal expressions produced by the construction above, i.e., they are not actually evaluated as vector fields belonging to a distribution. From this point of view, $\mathcal{LA}(\mathcal{X})$ is considered as a **Free Lie Algebra.** Our current problem is to enumerate a basis of this algebra, i.e., to get rid of redundant elements using only skew-symmetry and the Jacobi identity. Such a basis can be found via a Phillip Hall family.

The degree of an element X in **LA(X)** is denoted by degree(X) : this is the degree of the monomial defining X [17]. According to our notations, a **Phillip Hall family** (PH-family for short) of $\mathcal{LA}(\mathcal{X})$, is any totally ordered subset $(\mathcal{PH}, \prec)$ such that :

- If $X \in \mathcal{PH}$, $Y \in \mathcal{PH}$ and degree(X) < degree(Y) then $X \prec Y$;

. $\mathcal{X} \subset \mathcal{PH}$;

- $\mathcal{PH} \cap \mathcal{X}_2 = \{[X,Y], \quad X \prec Y\}$;

- An element $X \in \mathcal{LA}(\mathcal{X})$ with degree(X) $\geq$ 3 belongs to $\mathcal{PH}$ if and only if $X = [U, [V, W]]$ with U, V, W in $\mathcal{PH}$, $[V,W]$ in $\mathcal{PH}$, $V \preceq U \prec [V,W]$ and $V \prec W$.

The main property of a PH-family is that, taking skew-symmetry and the Jacobi identity into account, it yields a **basis** of the free Lie algebra $\mathcal{LA}(\mathcal{X})$ [6].

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

---

[16]The material used in this section comes from [6]. We want just to give a rough idea of the concept and of its pertinence with respect to our problem. Interested readers will find a more rigorous presentation in this reference.

[17] We use the word "degree" with two different meanings, according to whether we speak of a bracket or of a nonholonomic system. This may introduce some confusion, but both terms are already used in the literature (see for instance [57]).

4.2.2 The Algorithm

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

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

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

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

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

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

Example Part 1 : For a classical example [6] [55], take $\mathcal{X} = \{X_i, X_2\}$. The first 14 elements of the PH-family generated by the procedure (if it does not stop before) are:

X_1 $X_3 = [X_1, X_2]$ $X_4 = [X_1, [X_1, X_2]]$ $X_6 = [X_1, [X_1, [X_1, X_2]]]$ $X_9 = [X_1, [X_1, [X_1, [X_1, X_2]]]]$
X_2 $X_5 = [X_2, [X_1, X_2]]$ $X_7 = [X_2, [X_1, [X_1, X_2]]]$ $X_{10} = [X_2, [X_1, [X_1, [X_1, X_2]]]]$
                                    $X_8 = [X_2, [X_2, [X_1, X_2]]]$ $X_{11} = [X_2, [X_2, [X_1, [X_1, X_2]]]]$
                                                                        $X_{12} = [X_2, [X_2, [X_2, [X_1, X_2]]]]$
                                                                        $x_{13} = [[X_1, X_2], [X_1, [X_1, X_2]]]$
                                                                        $x_{14} = [[X_1, X_2], [X_2, [X_1, X_2]]]$

Example Part 2 : Consider now the controllability aspect. Replace X_1 and X_2 by the vector fields X_a and X_b defining the nonholonomic distribution of the 2-trailer convoy (see Section 3.2). The configuration space is a 5-dimensional manifold. Let c be a point of coordinates $(x,\ y,\ \theta,\ \varphi_1,\ \varphi_2)$. Yields :

$$X_1 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\sin\varphi_1 \\ \sin\varphi_1 - \cos\varphi_1\sin\varphi_2 \end{pmatrix} \qquad X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

The first elements of a PH-family are displayed in Example Part 1 (see Appendix 2 for the coordinates of the various vector fields). We can verify that the algorithm stops with $\{X_1, X_2, X_3, X_4, X_6)$ as a basis for every point c verifying $\varphi_1 \not\equiv \frac{\pi}{2} \bmod \pi$. The algorithm stops with $\{X_1, X_2, X_3, X_4, X_9\}$ for the remaining hyperplane[18].

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

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

The core of the algorithm is the construction of a PH-family. The dimension n of the manifold being a constant integer of our problem, the only tests needing a subroutine depending on n are (1) and (3). Their complexity is asymptotically negligible. Therefore the worst-case complexity of the algorithm is dominated by the complexity of building a PH-family. The relevant parameter is the value of i when the algorithm stops. Because of the test (2), our procedure for building a PH-family is not optimal[19]. But, here, we just want to find the minimal complexity of any algorithm that

---

[18]More precisely : $X_5 = X_1, \det\{X_1, X_2, X_3, X_4, X_6\} = -\cos(\varphi_1)$, $X_7 = 0$, $X_8 = -X_3$ and finally $\det\{X_1, X_2, X_3, X_4, X_9\} = -1 - \cos^2(\varphi_1)\cos(\varphi_2)$.

[19]It seems possible to define an optimal one. Maybe such a procedure already exists in the literature ?

builds a PH-family. Now, the complexity of computing all the elements of a set $\mathcal{H}_i$ is bounded below by the number of all the elements in $\mathcal{H}_j$, j ≤ i, and it has been proven that this number is

$$a(i) = \frac{1}{i} \sum_{d|i} \mu(d)(n - r)^{\frac{i}{d}}$$

where μ designates the Mobius function.

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

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

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

We have to prove this latter result. Indeed, the algorithm above clearly depends upon the basis $\mathcal{X}$ we chose for the distribution A. However, the concept of degree of nonholonomy does not. Now, it is a general result from the Lie Algebra Theory that $\bigcup_{j \leq i} \mathcal{H}_j$ constitutes a basis of the nilpotent free Lie algebra $\mathcal{LA}_i(\mathcal{X})$ defined by taking all the brackets of degrees less than i and by killing all the brackets of greater degrees. See [6] for details. Therefore, i does not depend on our choice of a basis $\mathcal{X}$ of A. It truly is the degree of nonholonomy that has been previously defined.

Example Part 3 : The degree of nonholonomy of the 2-trailer convoy is 4 at points whose coordinates (x, y, θ, φ_1, φ_2) verify $\varphi_1 \not\equiv \frac{\pi}{2}$ mod π. It is 5 elsewhere.

Summing up the results of this section :

The method we use for testing the controllability of a nonholonomic system at a point is at least exponential in the degree of nonholonomy at this point.

4.3 Growth Vector

This section introduces the growth vector of a controllable nonholonomic system at a point. It appears in [58]. This concept is a key one for the topological viewpoint that we will adopt in Section 6.

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

Example Part 4 : Let us recall that $\{X_1, X_2, X_3, X_4, X_6\}$ constitutes a basis for points whose coordinates (x, y, θ, φ_1, φ_2) verify $\varphi_1 \not\equiv \frac{\pi}{2}$ mod π, while $\{X_1, X_2, X_3, X_4, X_9\}$ works elsewhere. One can verify (by computing the dimension of the linear spaces for each level) that the growth vector at points verifying $\varphi_1 \not\equiv \frac{\pi}{2}$ mod π is $(2, 3, 4, 5)$, while it is $(2, 3, 4, 4, 5)$ elsewhere.

4.4 Singularities and Regular Systems

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

A filtration $\{\Delta_i\}$ is *regular* at a point c if the growth vector is constant in a neighborhood of c [58] [57]. This means that all the ranks of $\Delta_i(.)$ are constant in the neighborhood. Otherwise, the filtration is *singular* and the corresponding point is a singularity. Most of the problems we encounter when we try to define growth vectors and degrees of nonholonomy derive from the presence of singularities. By extension, we will say that a system is *regular* if the corresponding filtration is regular everywhere.

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

For regular systems, the degree of nonholonomy is a constant. It can also be shown (see [51]) that the growth vector is strictly increasing, so the procedure we designed always stops in that particular case.

4.5 Well-Controllability

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

As we have seen in Section 2, a general idea for devising a nonholonomic motion planner for controllable systems is to define a procedure that searchs for an admissible collision-free path, taking any collision-free path as a seed for the search[20].

Very recently Lafferiere and Sussmann proved that this principle is a general one. A collision-free path is first computed without taking the nonholonomic constraints into account. Lafferiere and Sussman's method [55] (see Section 6.3 for more details) roughly consists of expressing the first holonomic path into some "local coordinate system" (a precise definition will be given in Section 6.1); from these coordinates, because the system is controllable, the authors show that it is possible to **explicitely** define an admissible control (and then an admissible path) that locally steers the system from a given point (on the first path) to any other on the first path inside a given neighborhood. Because the planner has to work *a priori* everywhere, one has to define a procedure that guarantees to find a local coordinate system *everywhere.* The existence of such a coordinate system is a technical point essential for the method. It is solved by considering an *extended system* associated with the original one; this new system is obtained by adding virtual controls working on vector fields defined from a PH-family of the original system. Since the nonholonomic distribution A is *(n −* k)-dimensional, it seems *a priori* that *k* additional controls would suffice to make the system holonomic. In fact, in order to avoid singularities (understood as points where the transformation

---

[20][28] **pinpointed this method for the car-like robot, while** [32] **presents a planner using this principle for this case.**

matrix would be non invertible), one has generally to add more controls. Lafferriere and Sussmann note also that additional controls make easier the choice of a transformation matrix with a good condition number.

Let us illustrate this point using our example.

Example Part 6 : Recalling Example Part 2, a local coordinate system defined from $\{X_1, X_2, X_3, X_4, X_6\}$ will encounter singularities. Following Lafferiere and Sussmann's method, a possible extended control system is defined by $\{X_1, X_2, X_3, X_4, X_6, X_9\}$; in the process, four controls are added to the original ones. The previous results show that it is everywhere possible to choose in this family a basis that spans $\mathbf{R}^5$.

Now, consider the following family :

$$
\begin{array}{lll}
& U_0 = X_1 & V_0 = X_2 \\
Y_1 = [X_1, X_2] & U_1 = \cos \varphi_1 X_1 + \sin \varphi_1 Y_1 & V_1 = \sin \varphi_1 X_1 - \cos \varphi_1 Y_1 \\
Y_2 = [U_1, V_1] & U_2 = \cos \varphi_2 U_1 + \sin \varphi_2 Y_2 & V_2 = \sin \varphi_2 U_1 - \cos \varphi_2 Y_2 \\
Y_3 = [U_2, V_2] & &
\end{array}
$$

It is easy to check (see Appendix 3 for the general case) that the determinant of $\{V_0, V_1, V_2, U_2, Y_3\}$ is equal to 1. Therefore $\{V_0, V_1, V_2, U_2, Y_3\}$ spans $\mathbf{R}^5$ *everywhere*[21]. According to the previous comments, *we* can define a *minimal* extended system that never meets with any singularity. Moreover, the transformation matrix has a good condition number. We introduce the concept of a *well-controllable* system.

Definition 1 : An n-dimensional nonholonomic system defined by a distribution A is well-*controllable,* if there exists a basis of *n* vectors fields in the Control Lie Algebra LA(A) such that the determinant of the basis is constant.

Obviously well-controllability implies controllability. The converse does not hold. Indeed, as we mentionned in Section 4.1, a system can be controllable while the local degrees of nonholonomy are unbounded. This means that the filtration $\{\Delta_i\}$ stabilizes locally, but not globally. In this case, it is impossible to define a basis verifying the conditions of our definition.

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

Definition 2 : Let $\mathcal{B}$ be a basis of the control Lie Algebra verifying the conditions of Definition 1. The *degree of well-controllability* of the system is the maximum degree of all the elements of a.

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

---

[21] Be careful : the degree of Y_3 equals 8, when it is viewed as a polynomial function with indeterminates X_1 and X_2 in $\mathcal{LA}(\{X_1, X_2\})$.

Example Part 7 : The 2-trailer convoy is a well-controllable nonholonomic system. Its degree of nonholonomy is 5, while its degree of well-controllability is at most 8.

4.6 **Geometric Models and Singularities : some Examples**

Singularities typically come from the geometry of the system, though in some cases a bad choice of basis might create artificial singularities. Let us illustrate this point with three examples.

Consider once again the case of the l-trailer convoy. It appears (see Appendix 2) that this is a regular, well-controllable system of degree 3; its growth vector is $(2,3,4)$ everywhere. Now consider the same convoy, but with another hooking system. Instead of hooking the trailer exactly at the middle of the rear wheels, we hook it to a point behind the rear axle (see Figure 2)[22]. Solving the equations yields the basis:

$$
\begin{pmatrix} a' \cos \theta \\ a' \sin \theta \\ 0 \\ - \sin \varphi \end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix} (a' + a.\cos \varphi) \cos \theta \\ (a' + a.\cos \varphi) \sin \theta \\ \sin \varphi \\ 0 \end{pmatrix} .
$$

If we consider this basis only, we find a growth vector of $(2, 3, 4)$ at points verifying $\sin \varphi \neq 0$. The points verifying $\sin \varphi = 0$ seem to constitute a singularity. But, checking our provisional basis, we find that the vector fields are collinear at these points, though the distribution is regular at these points too. It is just awkward to find two vector fields which stay independent everywhere. For the record, any vector field of the form

$$
X = k \begin{pmatrix} a' \cos \theta \\ a' \sin \theta \\ 0 \\ - \sin \varphi \end{pmatrix} + l \begin{pmatrix} (a' + a.\cos \varphi) \cos \theta \\ (a' + a.\cos \varphi) \sin \theta \\ \sin \varphi \\ 0 \end{pmatrix} + m \begin{pmatrix} 0 \\ 0 \\ a' \\ a \cos \varphi + a' \end{pmatrix}
$$

is a valid vector field. With some ingenuity, for any value of a and a', it is possible to obtain a global basis.

Now let us pinpoint a stranger case. We have seen that the 3-body convoy we study is controllable with degree 5 . Appendix 2 shows that the 2-body convoy is controllable with degree 3. The l-body convoy (i.e., the unicycle) is controllable with degree 2. Because of the singularity we mentioned, there is no 4th degree. What is the fundamental difference between one trailer and two trailers ? Furthermore, we prove in Section 5 that a n-body convoy is well-controllable and that its nonholonomy degree is at most 2". What is its precise nonholonomy degree ? What kind of singularities do we stumble upon ?

Finally, our third example doesn't model any robotics systems whatsoever[23] . Rather, it tries to capture the flavor of problems encountered in practice without any tedious computations: for

---

[22][31] gives a constructive proof of controllability for this system.
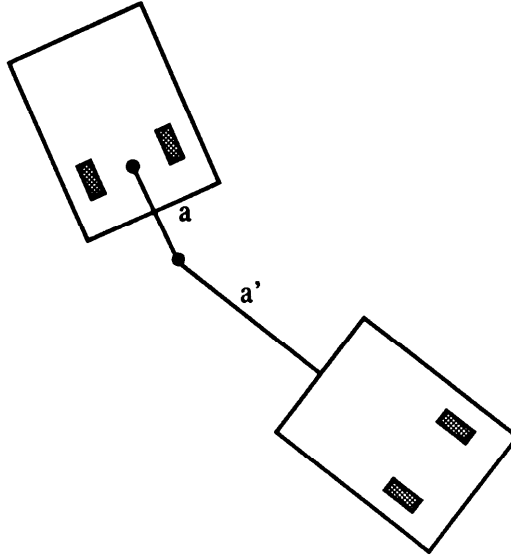[23]This example was suggested by Marc Espie.

Figure 2: Another l-trailer car system

instance, the example of the 2-trailer system is quite similar to this one. Let us work in Euclidean space (x, y, z). Choose an integer n and consider the following vector fields :

$$X = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad Y_0 = \begin{pmatrix} 1 + z^n \\ 1 + z^{2n} \\ 0 \end{pmatrix}$$

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

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

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

This is the typical case where our Algorithm will not terminate. A finer study of the problem would be a tremendous help.

4.7 Nilpotent and Nilpotentizable Systems

We have seen that the controllability testing procedure does not necessarily terminate in the general case. Notwithstanding, consider the following special case.

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

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

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

Such systems are called *nilpotent of order k* *(see* [6] for a general definition of the concept in the Lie Algebra framework).

Example Part 8 : In our example, we may verify (see Appendix 2) that $[X_2, [X_2, X_1]] = -X_1$. Set $ad_X(Y) = [X,Y]$. Then[24]: $ad_{X_2}{}^{2m}(X_1) = (-1)" \; X_1$. The system is not nilpotent.

In some cases, a non-nilpotent system can be transformed into a nilpotent one via a linear change of controls called a *feedback transformation.* Quite logically, such systems are called *feedback nilpotentizable.* [55] gives some examples of feedback nilpotentizations (e.g., the unicycle, a car-like system and a car-like system with a trailer). See also [18] for sufficient conditions for a system to be nilpotentizable.

Conjecture : Nilpotent and nilpotentizable systems are well-controllable.

The study of this conjecture, currently in progress, should help the quest for a general test of well-controllability.

4.8 **Triangular Systems**

The concept of a triangular system is used by Murray and Sastry in [39]. A system is *triangular* if it can be defined as :

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

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

Because of the triangular form, there exists simple sinusoidal control that may be used for generating motions affecting the ith set of coordinates while leaving the previous sets of coordinates unchanged. It is possible to use these sinusoidal controls for planning trajectories (see [39] for details).

Even if a system is not triangular, it may be possible to transform it into a triangular one by feedback transformations. Recent work [37] shows that a regular nilpotent controllable system can . be t riangularized.

---

[24] This example appears in [55] for the unicycle and the car-like robot, i.e., systems equivalent to our current system without trailers (see Section *4).*

26

Conjecture : Triangularizable systems are well-controllable.

The next section will mention as an *aparte* that the multibody car system we consider along this paper is triangularizable.

5 The Multibody Car System is Well-Controllable

Let us return now to the general case of our convoy with n trailers. We know a basis $\{X_a, X_b\}$ of the distribution A of the system (see Section 3.2) and we want to prove the existence of a family of fields in LA(A) that spans the full tangent space when applied anywhere. To solve this problem, the only difficulty is to find a good family. Moreover, a minimal family that works everywhere (*cf* the concept of well-controllability) is highly desirable.

Since the number of trailers is not fixed, we have to compute the general form of the *(n + 3)* coordinates of the vector fields we use.

Recall the form of X_a and X_b :

$$X_a = \begin{pmatrix} \cos \theta \\ \sin e \\ 0 \\ -\sin \varphi_1 \\ \sin \varphi_1 - \cos \varphi_1 \sin \varphi_2 \\ \cdot \\ (\sin \varphi_{i-1} - \cos \varphi_{i-1} \sin \varphi_i) \prod_{j=1}^{i-2} \cos \varphi_j \\ \\ (\sin \varphi_{n-1} - \cos \varphi_{n-1} \sin \varphi_n) \prod_{j=1}^{n-2} \cos \varphi_j \end{pmatrix} \qquad X_b = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

The vector fields of the basis are obtained through the use of the same combinations as in Section 4.6 (Example Part 5).

We build iteratively three brackets of degree 2^i from brackets of degree 2^{i-1} :

Step						
0			Y_0	=	X_a	Z_0	=	X_b
1	X_1	$= [Y_0, Z_0]$	Y_1	=	$\cos \varphi_1 Y_0 + \sin \varphi_1 X_1$	Z_1	=	$\sin \varphi_1 Y_0 - \cos \varphi_1 X_1$
2	X_2	$= [Y_1, Z_1]$	Y_2	=	$\cos \varphi_2 Y_1 + \sin \varphi_2 X_2$	Z_2	=	$\sin \varphi_2 Y_1 - \cos \varphi_2 X_2$
$\vdots$	$\vdots$		$\vdots$			$\vdots$		
i	X_i	$= [Y_{i-1}, Z_{i-1}]$	Y_i	=	$\cos \varphi_i Y_{i-1} + \sin \varphi_i X_i$	Z_i	=	$\sin \varphi_i Y_{i-1} - \cos \varphi_i X_i$
$\vdots$	$\vdots$		$\vdots$			$\vdots$		

The general form of the vector fields coordinates is given by a lemma (Lemma 1, in Appendix 3). It remains for us to choose $n + 3$ vector fields that constitute a basis of $\mathbf{R}^{n+3}$.

Lemma 2 : For any point c, $\{Z_0, Z_1, Z_2, \ldots, Z_n, Y_n, X_{n+1}\}$ (c) spans $\mathbf{R}^{n+3}$.

Proof : We just have to exhibit the following determinant :

$$
\begin{vmatrix}
0 & \beta & \beta & \beta & \ldots & \beta & \cos\alpha & -\sin\alpha \\
0 & \beta & \beta & \beta & \ldots & \beta & \sin\alpha & \cos\alpha \\
1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & \ldots & 0 & 0 & 0 \\
& & & \vdots & & \ddots & & \vdots \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0
\end{vmatrix}
$$

where a $= \left(\theta - \sum_{i=1}^{n} \varphi_i\right)$ (we do not need to express β). It appears clearly that this determinant equals (-1)". Q.E.D.

Finally, since the determinant does not vanish anywhere, we can conclude :

Property : The multibody car systems defined in Section 3.3 are well-controllable, with degree at most 2^{n+1}.

... and the driver of our luggage carrier can play rambling through the airport in search of a parking place ! An estimate of the time he needs to park his vehicle is the subject of the following section.

Remark: We have seen that such a system is not nilpotent. Because of the presence of singularities, it is not regular either. Nonetheless, the form we obtained for the determinant clearly shows that such a system is triangularizable.

6 The Complete Problem

In Section 2 we saw that the existence of an admissible collision-free trajectory for a controllable nonholonomic system is characterized by the existence of a collision-free trajectory for the associated holonomic system. Planning a trajectory can thus be achieved through the following steps:

- using a geometric planner for finding a trajectory without taking into account nonholonomic constraints[25] , and then

- steering the system as close by as possible to this trajectory along an admissible trajectory.

Such a general strategy has been refined into two different approaches that we will examine in Sections 6.3 [55] and 6.4 [33]. The first one uses a constructive proof of local controllability founded on fundamental tools of differential geometry introduced in the next section (Section 6.1). The second one uses an explicit form for canonical feasible trajectories (e.g., shortest trajectories).

Both approaches come up against the following key problem : how to guarantee that the admissible trajectory lies as close to the steering trajectory as the obstacles require ? The topological nature of this question is discussed in Section 6.2.

6.1   From Vector Fields to Trajectories

In this subsection we will investigate the delicate problem of finding trajectories that are compatible with our nonholonomic constraints. We need some precise concepts of differential geometry, which are thoroughly studied in [50] [57] [51] [52].

Let us first take the very simple example of an Euclidean plane. At any point, the tangent space is a two-dimensional Euclidean vector space, though one should not confuse the tangent space with the manifold itself. Pictorially, just view the Euclidean plane as a table and the tangent space at a point as a sheet of paper glued to the table at this very precise point. Considering the tangent space at another point just involves sliding the sheet and gluing it at the other point. One can then take a distinguished system of coordinates in these tangent spaces, namely the constant vectors X and Y. These vectors induce a system of coordinates on the manifold in a straightforward way. Starting from a given point (the origin), just move in the X direction for a given time, say a, then in the Y direction for a time b. You have reached the point of Cartesian coordinates a and b. There are some alternate ways to achieve that : you could have first followed Y for a time b, then X for a time a or, with self-assurance, have directly taken the direction of choice, the straight line along $aX + bY$. We can deduce some interesting facts from these simple manipulations. First, each of these procedures gives rise to a system of coordinates. That means that we have a one-to-one correspondence: every point is defined by its coordinates and reciprocally. That also means that the correspondence is smooth, two neighboring points have neighboring coordinates and the whole scheme is thoroughly unsurprising.

---

[25] Finding such a trajectory corresponds to the classical Piano Mover problem. This problem is decidable. From a theoretical point of view, general algorithms exist in the literature. Notwithstanding, at this time, there is no general software that runs efficiently in practice. This is due to the intrinsic combinatorial complexity of the problem.

We are now going to use the same scheme for our purposes. The main difference is that our vector fields won't be constant any longer, and won't even need to be globally defined. We could say that, instead of dealing with a flat Euclidean space, we will now work in a skewed space, where vector fields are allowed to behave horribly. But, nonetheless, the same constructions will work locally and give rise to some systems of coordinates-these are no longer equivalent.

Choose a point p in our manifold and a vector field X defined around this point. There is exactly one trajectory $\gamma(t)$ starting at this point and following X. In a formal notation, it verifies $\gamma(0) = P$ and $\dot{\gamma}(t) = X_{\gamma(t)}$.

One defines the exponential of X (denoted by e^X) to be the point y(l). This gives a correspondence between the space of vector fields and a neighborhood of p. It is obvious that one has $e^{tX} = \gamma(t)$, namely this definition doesn't describe a peculiar point of a trajectory but, more accurately, links every point of the trajectory to a specific vector field. Let us translate our previous example to this formalism. Following $aX + bY$ for a given time (the unit time) simply means to take e^{aX+bY}. Following X for a time a amounts to following aX for the unit time, that is taking e^{aX}, and following Y for a time b is the same as taking e^{bY}. This is still a slightly different point of view : instead of considering the exponential to define a specific point of a trajectory with regard to an origin point p, we understand it as describing a motion from a point to another on a given trajectory. Thus, starting at the origin o, following aX for a given time, then bY leaves us at the point $e^{bY} \cdot e^{aX} \cdot$ o Therefore the exponential of a vector field X appears as an operation on the manifold, meaning "slide from the given point along the vector field X for unit time."

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

Assume that $X_1, \ldots, X_n$ are vector fields defined in a neighborhood U of a point p such that at each point of U, $X_1, \ldots, X_n$ constitutes a basis of the tangent space. Then there is a smaller neighborhood V of p on which the correspondences *(al, . . . , a,)* $\mapsto e^{a_1 X_1 + \cdots + a_n X_n} \cdot p$ and $(a_1, \ldots, a_n) \mapsto e^{a_n X_n} \ldots e^{a_1 X_1} \bullet p$ are two coordinate systems, called the first and the second normal coordinate system associated to {Xi,. . . , X,}.

The Campbell-Haussdorf-Baker-Dynkin formula states precisely the difference between the two systems :

For a sufficiently small t, one has : $e^{tX} \cdot e^{tY} = e^{tX+tY-\frac{1}{2}t^2[X,Y]+t^2\epsilon(t)}$, where $\epsilon(t) \to 0$ when t $\to$ 0.

Actually, the whole Campbell-Haussdorf-Baker-Dynkin formula as proved in [57] gives an explicit form for the ϵ function. More precisely, ϵ yields a formal series whose coefficients lie in $\mathcal{LA}(\{X, Y\})$: the coefficient of t^k is a combination of brackets of degree k. In the case of a nilpotent system of order **k,** since brackets of degrees greater than **k** vanish, the Campbell-Haussdorf-Baker-Dynkin gives an *exact* development of the exponential. This property is used in the Lafferiere and Sussmann's planner presented below.

Roughly speaking, the Campbell-Haussdorf-Baker-Dynkin formula tells us how a nonholonomic system can reach *any* point in a neighborhood of a starting point. This formula is the hard core of the local controllability concept. It yields a method for *explicitly computing trajectories* in a neighborhood of a point.

Now we take a closer look at the problem of obstacle avoidance.

6.2 Obstacle Avoidance : the Topological Question

6.2.1 What Kinds of Topologies ? An Informal Statement

Let us come back to the sources. Motion planning in Robotics deals with obstacle avoidance. Real obstacles get transformed into "obstacles" in the configuration space. The Hausdorff metric[26] on the bodies, which are closed compact subsets, in the environment (that is to say, 3-dimensional Euclidean space) induces a metric in the configuration space. Therefore, the reference open sets in the configuration space are the open sets in the topology induced by the Hausdorff metric in Euclidean space. This is the topology needed for solving placement problems[27]. A path appears as a continuous function from a closed interval of $\mathbf{R}$ to the configuration space equipped with this topology.

In some cases, the very act of considering motions can lead to a finer topology. If we introduce the distance induced by the best trajectory(ies) between two points with respect to a given cost (length, energy, time taken, etc), differential considerations take the scene. Consider the case of energy for instance. For holonomic systems, since every smooth path in the configuration space is an admissible trajectory, this cost induces a natural Riemannian distance. In that case, the induced topology remains the same.

For general nonholonomic systems, there may exists points at an infinite distance of each other. The non-holonomic constraints partition the configuration space into disconnected submanifolds, and the resulting topology has little ressemblance to the natural one. However, for controllable nonholonomic systems, any two points can be joined by an admissible trajectory; considering the best trajectory leads once again to a well-defined metric. This is the origin of sub-Riemannian geometry. Recent contributions show that Riemannian and sub-Riemannian metrics are equivalent. Therefore both topologies are the same. The next section state these results more thoroughly.

6.2.2 Of Sub-Riemannian Metrics, Shortest Paths and Geodesics

This section makes use of the ideas given in [58] for the case of regular systems. Consider a controllable nonholonomic system (i.e., a completely nonholonomic one) defined on a n-dimensional

---

[26] A Hausdorff metric can be defined for any space equipped with a distance d. It yields the following topology on compact subsets :
$$d_H(A, B) = inf_{x \in X}(sup_{y \in Y} d(x, Y)).$$

[27] See the problem of cloth or leather cutting as an example of such a problem in the context of Computational Geometry.

manifold **CM** (for "Configuration Manifold") by a distribution A. The nonholonomic metric[28][58] is defined by

$$\rho_\Delta(c, c') = \inf_{\gamma \in S(c,c')} \int_0^1 \langle \dot\gamma(t), \dot\gamma(t) \rangle^{\frac{1}{2}} dt,$$

where

$$S(c, \text{c'}) = \left\{ \gamma \mid [0, 1] \to CM, \; \gamma(0) = c, \; \gamma(1) = \text{c'}, \; \dot\gamma \in A \right\} .$$

In that setting, geodesics are admissible trajectories that locally solve the variational problem.

The proof of the equivalence of the Riemannian and the sub-Riemannian metric resides in a two-sided estimate of the size of sub-Riemannian balls. Denote by B,(c) the sub-Riemannian c-ball, i.e., the set of points reachable from c by an admissible trajectory of length less than ϵ. Assume that the system is **regular** at c (see Section 4.8). Let {c;} be the local coordinate system defined in Section 6.1. Now consider the parallelepiped $P_{\alpha,\epsilon}(c) = \left\{ c' \in CM \mid |c_i(c')| \le \alpha\epsilon^{\varphi(i)} \right\}$ where $\varphi(i)$ is derived from the growth vector $\{n_1, \ldots, n_{q_c}\}$ of A at c as $\varphi(i) = j$ for $n_{j-1} < i \le n_j$.[29] A two-sided-estimate of **B,(c)** is given by the **parallelepiped theorem** [58][30] : there are positive constants $\alpha_1, \alpha_2, \epsilon_0$, such that for $\epsilon < \epsilon_0$,

$$P_{\alpha_1,\epsilon}(c) \subset B_\epsilon(c) \subset P_{\alpha_2,\epsilon}(c).$$

Therefore, for regular systems, Riemannian and sub-Riemannian topologies are the same.

Remark : The parallelepiped theorem holds only at regular points. There are some technical problems at singular points (e.g., as at the singularities appearing in the 2-body system). It seems possible to extend the proofs (using a local coordinate system, i.e., valid everywhere in a neighborhood of a regular point) to well-controllable systems by using a local coordinate system holding for singular points as well as for regular points (such a system exists by definition). Nevertheless, as always with singularities, some care will have to be taken.

Strange phenomena appear in this sub-Riemannian geometry framework. We know that, in general, if shortest paths are geodesics, geodesics are not necessarily shortest paths-indeed, they only minimize length **locally.** One of the main features of Riemannian geometry is that, locally, geodesics **are** shortest paths. Consider the e-ball B,(c) above, and S,(c) its boundary sphere. In Riemannian geometry, for ϵ sufficiently small, the sphere S_ϵ is in one to one correspondence with the ends of geodesics of length ϵ (the so-called **wave front).** No similar property does hold for nonholonomic systems. [58] holds two drawings illustrating the strange relationship between the spheres and the wave fronts in the Heisenberg group case.

---

[28] This metric is also known as a singular [7], a *Carnot-Caratheodory* [36], or a *sub-Riemannian* [51] metric.

[29] **Example** Part 9 : For the Z-trailer convoy at $c = (0, 0, 0, 0, 0)$, the growth vector is $(2, 3, 4, 5)$ (see Example Part 4). Therefore :

$$P_{\alpha,\epsilon}(c) = \left\{ (x, y, \theta, \varphi_1, \varphi_2) \mid |x| \le \alpha\epsilon, \quad |y| \le \alpha\epsilon, \quad |\theta| \le \alpha\epsilon^2, \quad |\varphi_1| \le \alpha\epsilon^3, \quad |\varphi_2| \le \alpha\epsilon^4 \right\}$$

[30] Bibliographic note : the proof of this theorem does not appear in [58]. A proof using different terminology appears in [51].

Another strange phenomenon can be illustrated by the study of the car-like system[31]. In this particular case, the shortest paths have an explicit form (see the following section). Figure 3 is a rendering of one of the corresponding spheres. Just notice that this sphere is not smooth and look at the planes cutting the smooth part.

6.2.3 Geodesics and Shortest Paths : Elementary Computational Aspects

The classical way for computing geodesics is to use the maximum principle of Pontryagin [43]. This is a powerful tool from classical Optimal Control Theory, which provides necessary conditions for the existence of an optimal control. The bang-bang principle is its direct consequence [1]. Under some hypotheses, this principle gives the form of the optimal controls, if they exist. It has been applied in [19] for Hilare-like robots (see Section 3.3.2) : in this case, geodesics are piecewise clothoids or anticlothoids. Using the same ideas, we may verify that the geodesics for car-like robots (see Section 3.3.3) are piecewise arcs of circle or straight lines.

How to compute shortest paths is a much more difficult problem, even when an explicit form for the geodesics (local shortest paths) is known. The problem is basically a combinatorial one : how to piece smooth geodesic parts together in order to produce a shortest path ? As far as the author knows, this question has only been answered for the car-like robot by Reeds and Shepp [44]. They extend the work of Dubins on the form of smooth shortest paths [14] and they establish precisely which combinations of arcs of circle and straight line segments can produce shortest paths. Since the number of used combinations is finite, this gives birth to an efficient method to compute shortest paths.

6.3 A Planner Using Philipp Hall Coordinate Systems

This section is only a sketch of the general approach developed by Lafferiere and Sussmann in [55]. First, they study nilpotent and nilpotentizable systems. For such systems we have seen that it is possible to compute a basis $\mathcal{B}$ of the Control Lie Algebra LA(A) from a Philipp Hall family. Their method assumes that a holonomic trajectory γ is given. If we express locally this trajectory on $\mathcal{B}$, i.e., if we write the tangent vector $\dot{\gamma}(t)$ as a linear combination of vectors in $\mathcal{B}(\gamma(t))$, the resulting coefficients define a control that steers the holonomic system along γ. Using the Campbell-Haussdorf-Baker-Dynkin formula, it is then possible to compute an admissible control u for the nonholonomic system that steers the system *exactly* to the goal (indeed, since all the brackets vanish after a given level **k,** the Campbell-Haussdorf-Baker-Dynkin formula gives an *exact* development of the exponential on brackets of degree less than **k,** so the synthetized trajectory ends exactly at the same point.

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

---

[31]If we allow further inequality constraints on the controls (see Remark in Section 3.3.3), [5] shows how to relate to the sub-Riemannian geometry framework.

In both cases, the nonholonomic trajectory is a local approximation of γ. In the presence of obstacles, the "gap" due to the approximation has to be estimated in order to avoid the obstacles. This point is not clearly mentioned in the preliminary version of [55]. In fact, their algorithm can be used as is by performing a recursive subdivision on the holonomic trajectory until all endpoints can be linked by *collision-free* trajectories synthetized by their algorithm above (this idea is also used in the second planner described below). It also seems that the parallelepiped theorem could be used to prove that the subdivision procedure always stops.

Remark : Lafferiere and Sussmann's strategy can be extended to well-controllable systems by replacing the Philip Hall local coordinate system by another one built from any basis working everywhere. This extension is currently under study.

6.4 A Planner Using Shortest Paths

The planner developed by Jacobs and Laumond [33] uses a shortest paths approach. The three steps of the algorithm are as follows :

- Plan a trajectory γ for the corresponding holonomic system. If one does not exist, then no feasible trajectory exists.

- Subdivide γ until all endpoints can be linked by a minimal length *collision-free* feasible trajectory.

- Run through an "optimization" routine to reduce the length of the trajectory.

The convergence of the algorithm is a consequence of the parallelepiped theorem. Indeed, consider a point c on γ and $\mathcal{N}_c$ a neighborhood of c included in the *collision-free* configuration space. The parallelepiped theorem guarantees that there is a neighborhood $\mathcal{N}_c'$ of c, such that for each point c' $\in \mathcal{N}_c'$ the shortest trajectory between c and c' lies in $\mathcal{N}_c$. Therefore since γ can be covered by a finite number of such neighborhoods $\mathcal{N}_c$, the subdivision procedure will stop.

This planner has been totally implemented in an exact version for the special case of a polygonal car-like robot. This means we had to implement a geometric planner for computing an exact representation of the collision-free configuration space. To do this, we used Avnaim and Boissonnat's algorithm [2]. We also implemented a procedure for computing the shortest trajectories in the absence of obstacles based upon Reeds and Shepp's work. Finally, we designed a fast collision checking procedure. Figure 4, an excerpt from [33], shows how the algorithm responds to the classical example of the parking problem. The three drawings give the trajectories produced by the three steps of the algorithm.

An advantage of such a strategy is that it optimizes locally the trajectory (in terms of the trajectory length). Indeed, the third step finds a quasi-optimal solution. Notice that finding the optimal one is a problem known to be very difficult, and computationally very complex.

The main drawback of this general strategy is that we need to compute the shortest paths (see Section 6.3.3). We are working on the Hilare-like system : at this time we know an explicit form for the geodesics [19], but not for the shortest paths.

Notwithstanding, using general mathematical techniques for proofs of controllability and complexity analysis, we have laid a theoretical basis for a study of systems of greater complexity. Essentially, the method consists in establishing a catalogue of canonical trajectories having the necessary topological properties, then in using them together with the subdivision motion planning technique. Clearly, the main question is : how to compute a sufficient set of canonical trajectories ? In the general case, the only way out seems to be the use of discretization techniques, but even in this case, the theoretical background is required to find a discretization fine enough to solve the problem. This latter aspect is currently under investigation.

6.5 **Complexity of the Complete Problem**

As discussed in Section 2, the *decision* part of the motion planning problem for controllable systems is equivalent to the decision problem for the associated holonomic system. Hence the complexity of this problem is a polynomial function of the *complexity* of the environment (i.e., the number of geometric primitives required to describe it), and the classical algorithms for the piano movers problem can be applied.

Notwithstanding, the complexity of the *complete* problem (i.e., actually producing a trajectory) is more difficult to grasp. As a rule, any measure of complexity for this problem ought to be bounded from below by the complexity of the solution, which in turn ought to account for the number of singular points (cusps, loops,. . .) it contains. Unfortunately, this number bears no relationship to the customary description of the input data used for the piano movers problem. It actually depends on the inner size of the free space, measured in the sub-Riemannian setting adapted to the problem.

Before giving a formal definition of the complexity of a trajectory, as it appears in [5], we will first introduce the concept for a continuous function $f\colon [a,\ b] \to$ R. A good measure of what we call the *geometric complexity* $GC(f)$ of such a function is the number of changes of variation. More precisely, this complexity is defined as the quotient of the total variation of the function (see [45]) by the amplitude of the function.

In the current context, our trajectories depend on the associated controls, which are continuous real valued functions. Consequently, we define the complexity of a trajectory according to the complexity of the associated controls. If a trajectory γ is defined by a control function u, its geometric complexity $GC(\gamma)$ is :

$$GC(\gamma) = \frac{\int_{\|\xi\|=1} GC(\langle u, \xi \rangle)\, d\xi}{\int_{\|\xi\|=1} d\xi}.$$

GC $((u, \xi))$ denoting the geometric complexity of the continuous real valued function (u, ξ), projection of u along the axis supporting ξ (see [5] for details).

This definition grasps every critical point of a trajectory, and so matches closely the output complexity of any motion planner. In the current context, it also includes the number of maneuvers as well as the number of loops and inflexion points.

This definition hints at some lower bounds for the complexity of complete motion planners. For instance [5] solves the case of the car-like robot. Given c and c' two configurations in the same

connected component of configuration space, we proceed to define ϵ. Choose an admissible path between c and c'. At a point q of that path, find the Euclidean diameter of the largest ball centered at q and wholly contained in the free configuration space. Take the infimum of this diameter for all points of the path. Then ϵ is defined as the supremum of this quantity over all possible paths linking c and c'.

Then the complexity of a trajectory of a car-like robot going between c and c' is proportional to $O(\epsilon^{-2})$.

Roughly speaking, this means that, in the car parking problem (as shown in Figure 4), the number of necessary maneuvers asymptotically varies as the square of the inverse of the margin of maneuver.

The proof of this result is based upon the Campbell-Haussdorf-Baker-Dynkin development of the exponential. A detailed study of the general case exhibits very close relationship between this complexity model and the degree of nonholonomy of the system. Everything depend upon the shape of the parallelepipeds derived from the growth vectors (Section *6.2.2)*. A general proof, currently under study, should lead to the following property :

Property : Given c, c' two configurations in the same arcwise-connected component of the configuration space. Define ϵ as the supremum, over all continuous paths from c to c', of the infimum, over all configurations q along that path, of the Euclidean diameter of the largest ball centered at q and wholly contained in the free configuration space. The problem of finding the trajectory for a *regular* controllable nonholonomic system of degree **k** between c and c' is in $O(\epsilon^{-k})$.

Conjecture : The problem is in $O(\epsilon^{-k})$ for well-controllable systems whose well-controllability degree equals **k**.

If our conjectures hold, this would mean that **our luggage carrier driver would have to do $0\left(\epsilon^{-(n+1)}\right)$ maneuvers in order to park his machine in the neighborhood of regular points, and in $O\left(\epsilon^{-2^{(n+1)}}\right)$ in the worst case . . .**

Acknowledgments

References

[1] M. Athans, and P.L. Falb. *Optimal Control.* McGraw-Hi& 1966.

[2] F. Avnaim, J.D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *IEEE International Conference on Robotics and Automation, Philadelphia (USA),* pages 1656-1661, April 1988.

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

[4] J. Barraquand and J.C. Latombe. Controllability of mobile robots with kinematic constraints. *Technical Report STAN-CS-90-1317,* Stanford University, June 1990.

[5] A. Bellaiche, J.P. Laumond and P. Jacobs Controllability of car-like robots and complexity of the motion planning problem with kinematic constraints. *Int. Symp. on Intelligent Robotics,* Bengalore, January 199 1.

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

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

[8] John Canny, Bruce Donald, John Reif, and Patrick Xavier. On the complexity of kinodynamic planning. In 29th *Symposium on the Foundations of Computer Science,* pages 306-316, White Plains, NY., October 1988.

[9] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. *6th ACM Symp. on Computational Geometry,* Berkeley, 1990.

[10] R. Chatila. Mobile Robot navigation: space modeling and decisional processes. In *Robotics Research : The Third International Symposium, 0. Faugeras and G. Giralt (Eds), MIT Press, Cambridge, Massachusetts, 1986.*

[11] P. Calou, W. Choi, J.C. Latombe, C. Lepape and M. Yim. Indoor automation with many mobile robots. In *IEEE IROS,* 1989.

[12] W.L. Chow. Uber systeme von lenearen partiellen Differetialgleichungen ester ordung. *Math. Ann.,* 117, pp 98-105, 1939.

[13] C. A. Desoer and Y. T. Wang. Linear-time-invariant servomechanism problem: A self-contained exposition. In C. T. Leondes, editor, *Control and Dynamic Systems,* pages 81-129. Academic, New York, 1980.

[14] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics,* 79:497–516,1957.

[15] S.J. Fortune and G.T. Wilfong. Planning constrained motions. In *ACM STOCS,* pages 445-459, Chicago IL, May 1988.

[16] G. Giralt, R. Chatila, and M. Vaisset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In *Robotics Research : The First International Symposium, M. Brady and R. P. Paul (Eds), MIT Press, Cambridge, Massachusetts, 1984.*

[17] Hermann and Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control,* 22(5), *1977.*

[18] H. Hermes, A. Lundell, and D. Sullivan. Nilpotent bases for distributions and control systems. *J of Differential Equations, 55, 385-400,* 1984.

[19] P. Jacobs, A. Rege, and J.P. Laumond. Non-holonomic motion planning for Hilare-like mobile robots. In *Int. Symp. on Intelligent Robotics,* Bengalore, January 1991.

[20] Paul Jacobs. Minimal length curvature constrained paths in the presence of obstacles. LAAS/CNRS Report 90042, Laboratoire d'Automatique et d'Analyse des Systemes, February 1990.

[21] Paul Jacobs and John Canny. Planning smooth paths for mobile robots. In *Proceedings of the 1989 International Conference on Robotics and Automation,* pages 2-7. IEEE, May 1989.

[22] Paul Jacobs and John Canny. Robust motion planning for mobile robots. In *Proceedings of the 1990 International Conference on Robotics and Automation,* Cincinnati, Ohio, May 1990. IEEE.

[23] Paul Jacobs, Greg Heinzinger, John Canny, and Brad Paden. Planning guaranteed near-time-optimal trajectories for a manipulator in a cluttered workspace. Tech. Rep. RAMP 89-15, University of California, Berkeley, Engineering Systems Research Center, September 1989.

[24] K. Kedem and M. Sharir. An efficient motion-planning algorithm for a convex polygonal object in two-dimensional polygonal space. *Discrete and Computational Geometry,* 5:43–75, 1990.

[25] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research,* 5(1):90–99, Spring 1986.

[26] B.H. Krogh and D. Feng. Dynamic generation of subgoals for autonomous mobile robots using local feedback information. *Transactions on Automatic Control,* 34(5):483–493, May 1989.

[27] J.C Latombe. Robot Motion Planning. *Kluwer Pub.,* to appear, 1990.

[28] J.P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In *Intelligent Autonomous Systems,* North Holland, 1987.

[29] J.P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *10th International Joint Conference on Artificial Intelligence (IJCAI), Milan (Italy), 1987.*

[30] J.P. Laumond, M. Taix, and P. Jacobs. A motion planner for car-like robots based on a mixed global/local approach. In *IEEE IROS,* July 1990.

[31] J.P. Laumond and T. Simeon. Motion planning for a two degrees of freedom mobile robot with towing. *LAAS/CNRS Report* 89148 , Toulouse, April 1989.

[32] J.P. Laumond, T. Simeon, R. Chatila, and G. Giralt. Trajectory planning and motion control for mobile robots. In *Geometry and Robotics, JD. Boissonnat and JP. Laumond (Eds),* pages 133-149. Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.

[33] P. Jacobs, J.P. Laumond, M. Taïx, and R. Murray. Fast and exact trajectory planning for mobile robots and other systems with non-holonomic constraints. Technical Report LAAS/CNRS 90318, September 1990.

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

[35] C. Lobry. Controlabilite des systkmes non linéaires. *SIAM J of Control,* 8:573–605, 1979.

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

[37] R. Montgomery, R.M. Murray, and S.S. Sastry. Triangulating differential equations on homogeneous nilpotent Lie groups. (work in progress).

[38] R.M. Murray and S.S. Sastry. *Mathematical Questions in Robotics,* chapter Mathematical Problems in Grasping and Manipulation by Multifingered Robot Hands. American Mathematical Society, January 1990.

[39] R.M. Murray and S.S. Sastry. Steering nonholonomic systems using sinuoids. *29th C.D.C., Honolulu, Dec. 1990*

[40] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robots via bi-directional approach. *IEEE International Conference on Robotics and Automation, 1990.*

[41] Colm Ó'Dúnlaing. Motion planning with inertial constraints. Robotics Report 73, New York University, July 1986.

[42] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. *IEEE International Conference on Robotics and Automation, 1990.*

[43] L. Pontryagin et *al. The Mathematical Theory of Optimal Processes.* L.S. Pontryagin Selected Works, Vol. 4, Gordon and Breach Sc. Pub., 1986.

[44] J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J Mathematics,* to appear.

[45] H.L. Royden. *Real Analysis.* The Macmillan Company, New York, 1968.

[46] E. Rimon and D.E. Koditschek. The construction of analytic diffeomorphisms for exact robot navigation on star worlds. In *Proceedings of the 1989 IEEE Int. Conference on Robotics and Automation,* pages 21-26, Scottsdale, May 1989.

[47] C. Samson and K. Ait-Abderrahim. Mobile Robot Control Part 1 : Feedback Control of a Nonholonomic Wheeled Cart in Cartesian Space. Technical Report 1288, INRIA, Sophia Antipolis, 1990.

[48] M. Sharir. Algorithmic motion planning in robotics. Technical Report 392, Courant Institute, New-York University, 1988.

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

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

[51] R.S. Strichartz. Sub-riemannian geometry. *J Differential Geometry, 24,* 221–263, 1986.

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

[53] H.J. Sussmann and Jurdjevic. Controllability of nonlinear systems. *J of Differential Equations,* 12, 1972.

[54] H.J. Sussmann. Lie brackets, real analyticity and geometric control. In *Differential Geometric Control Theory,* Brockett, Millman, Sussmann Eds, Birkhäuser, 1982.

[55] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift : a preliminary report. *Technical Report* SYCON-90-04, Rutgers Univ., June 1990.

[56] P. Tournassoud. Motion planning for a mobile robot with a kinematic constraint. In *Geometry and Robotics, JD. Boissonnat and JP. Laumond (Eds),* pages 150–171. Lecture Notes in Computer Science, Vol 391, Springer Verlag, 1989.

[57] V.S. Varadara'an. *Lie Groups, Lie Algebra, and their Representations,* Springer-Verlag, 1984.

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

[59] G.T. Wilfong. Motion planning for an autonomous vehicle. In *IEEE Int. Conf. on Robotics and Automation, 1988.*

[60] C.K. Yap. *Algorithmic and geometric aspects of Robotics,* chapter Algorithmic motion planning. J.T. Schwartz and C.K. Yap (eds), Lawrence Erlbaum Associates, London, 1987.

7 Appendix 1 : Related Work and Background

This section presents an overview of results recently obtained in constrained motion planning. They are complementary to those mentioned in the main paper, in the extent that they take a computational point of view into account.

7.1 Car-like and Trailer-like Robots

[28] envisaged the controllability of a car-like robot. The main result is that the existence of a collision free trajectory for such a system is characterized by the existence of a connected component in the free configuration space. The constructive proof of this result has led to the implementation of a planner [32]. This planner uses an approximate representation of the free configuration space decomposed into parallelepipeds.

The planner described in [3] uses a discretization of the phase space, the adjacency relation between two cells taking kinematic constraints into account. Collision tests are obtained from a bitmap representation of the 2D environment. The search algorithm applies a best-first search strategy whose cost function is the number of maneuvers. The planner then produces paths with a minimal (for the discretized representation) number of maneuvers.

Notice the heuristic approach developed in [56] : the 2D environment is decomposed into a set of corridors in which a specific technique for planning smooth paths is applied. Finally [59] proposes an algorithm that produces a motion with the minimum number of turns for a set of lanes extracted from the environment.

The case of trailer-like robots was attacked by Laumond and Siméon [31]; they established the controllability of such systems by applying the Lie Algebra formalism briefly recalled in the introduction. They also give a second constructive proof, but this one does not lead to an efficient algorithm.

The Barraquand and Latombe planner cited above also applies to the case of a robot with trailers. Their planner is thus the first to produce paths for such a system.

7.2 Smooth Paths

Planning smooth, maneuver-free, paths for a mobile robot appears to be a more difficult problem than when we allow maneuvers. In fact, there is no comparable controllability result. It is indeed possible that a path exists for a holonomic system, and yet no smooth path exists. The specific problem was first addressed by Dubins [14], who gives the form of the shortest bounded curvature path in the absence of obstacles. The problem of obstacle avoidance appears more recently in [28]. In [29], the environments consist of closed curves which are not necessarily polygons. Unfortunately, the method presented there is not guaranteed to find a path. In Fortune and Wilfong [15], a decision algorithm is given to decide if a path exists under given conditions. The algorithm is exact, but does not generate the path in question. This algorithm runs in exponential time and space. The algorithm in [21] is a provably-good approximate algorithm solving the problem. It depends on a simplification of the set of smooth trajectories to a sufficient subset of canonical trajectories.

7.3 Shortest Paths

The study of shortest paths for constrained systems began with the seminal work of Dubins [14], who found the form of minimal length smooth paths with bounded average curvature for the nonholonomic system we study in this work. He established his result in the case that there are no obstacles. The algorithm presented in [21] is a provably good approximate algorithm, which depends upon an extension of the results of Dubins to the case where there are obstacles in the environment. This extension has been presented in a rigorous manner in [20]. When there are obstacles, minimal-length smooth paths consist of paths of the forms given by Dubins, alternating with parts of the obstacle boundaries. Finally, the same proof holds for a polygonal robot moving in a polygonal environment. However, in this case the portions of the obstacle boundaries are replaced by logarithmic curves which we call generalized tractrices. These curves are those that maintain contact between an obstacle edge and a robot vertex while satisfying nonholonomic kinematic constraints [56]. A study is currently in progress to exploit this characterization of the minimal-length paths and develop a provably good algorithm finding smooth paths for a polygonal robot.

Recently Reeds and Shepp extended the work of Dubins to piecewise smooth paths with bounded curvature, alas without obstacles [44]. That is, they allow maneuvers and cusps along the path.

Essentially they proceed to show that any path holding more than two cusps is reducible to a path with at most two cusps, no longer and possibly shorter. It is clear that between cusps, the path must be of the form given by Dubins. Then they discard some of the allowable curves by the use of a homotopy argument.

Because, in this paper, we have concentrated on the case in which maneuvers are allowed, we have used these curves as the basis for the development of our algorithms.

7.4 Time-Optimal Paths

There has also been some work on planning time-optimal trajectories for systems with constrained accelerations. Fundamental work in trajectory planning with dynamic constraints is presented in [41]. That paper introduces the idea of planning trajectories for a particle with constraints on its acceleration. Unfortunately, the analysis is restricted to the case of one dimension motion. In [8] the same problem is addressed in the multiple dimension case. That work discusses finding a near-time-optimal safe trajectory for a moving particle subject to uniform L_∞ acceleration bounds on each axis. Unfortunately, the constraints on accelerations which are allowed in such approaches, while they may be state-dependent [23], do not apply to the systems we study here. More recently, [9] presents the first exact algorithm for time-optimal kinodynamic motion planning in the two-dimensional case.

7.5 Control-Oriented Approaches

There are sound arguments for applying ideas from Control Theory to motion planning. Control theoreticians have long been interested in finding controllers which would be robust to modeling errors, would reject disturbances and cope with parameter fluctuations (for example, see [13]).

There is a small body of literature on the obstacle avoidance problem using some ideas from Control Theory. Most of this work centers on the technique of ***potential fields*** introduced by Khatib [25]. In such problems, a potential function is set up so that the system will travel to the goal state by moving in the direction of the negative gradient. Obstacles are avoided by making them areas of high potential. This work suffers from being a local method that can abort before reaching the desired goal state. However, an advantage is that these techniques are inherently robust. Extensions have been proposed by Rimon and Koditschek [46] in an attempt to use this approach for global path planning. Krogh and Feng [26] have also proposed a method of global planning in which the location of a subgoal is continuously changed to lead the system around local minima generated by a set of convex obstacles to the actual goal.

8 Appendix 2 : The 2- and 3-Body Car Systems

This appendix presents the listings in **Mathematica** of computations of the various vector fields corresponding to the examples of the **2-** and **3-Body** Systems respectively.

```
(*
 * A definition for Jacobians and Lie derivatives
 *)

Clear[Jac,Lie]

Jac [x_, vect_]  := Table[D[x[[i]], vect[[j]]], {i, Length[x]},{j, Length[vect]}]

Lie[x_, y,, vect_] := Jac[y, vect].x - Jac[x, vect].y

(*
 * The 2-body system
 *)

vect = {x, y, theta, phi)
(* a "simple" lie function, that simplifies brackets along the way *)
<<Trigonometry.m
slie [x_, y_]  := Map[TrigCanonical, ExpandAll[Lie[x, y, vect]]]


(* Phillip Hall Family H1 *)
x1 = {Cos[theta], Sin[theta], 0, - Sin[phi]}
x2 = (0, 0, 1, 1)

(* Phillip Hall Family H2 *)
x3 = slie[x1, x2]
(* Phillip Hall Family H3 *)
x4 = slie[x1, x3]
x5 = slie[x2, x31
```

This yields the following results:

$$x_1 = \{\cos(\theta), \sin(\theta), 0, -\sin(v)\}$$
$$x2 = \{0, 0, 1, 1\}$$
$$x_3 = \{\sin(e), -\cos(\theta), 0, \cos(\varphi)\}$$
$$x_4 = \{0, 0, 0, 1\}$$

$$\det(x1,\ x2,\ x3, x4) =\ 1$$

```
(*
 * The 3-body system
 *)

vect = (x, y, theta, phi1, phi2}

(* Phillip Hall Family H1 *)
x1 = {Cos[theta], Sin[theta], 0, - Sin[phi1], Sin[phi1] - Cos[phi1] Sin[phi2]}
x2 = {0, 0, 1, 1, 0}
(* Phillip Hall Family H2 *)
x3 = slie[x1, x21
(* Phillip Hall Family H3 *)
x4 = slie[x1, x3]
x5 = slie[x2, x33
(* Phillip Hall Family H4 *)
x6 = slie[x1, x4]
x7 = slie[x2, x4]
x8 = slie[x2, x5]
(* Phillip Hall Family H5 *)
x9 = slie[x1, x6]
 (* an actual controllability algorithm would stop here *)
x10 = slie[x2, x6]
x11 = slie[x2, x7]
x12 = slie[x2, x8]
x13 = slie[x3, x43
x14 = slie[x3, x53
```

In that case, we obtain the following vector fields.

$$
\begin{aligned}
x1 &= \{\cos(\theta), \sin(\theta), 0, -\sin(\varphi_1), \sin(\varphi_1) - \cos(\varphi_1)\sin(\varphi_2)\}\\
x2 &= \{0,0,1,1,0\}\\
x3 &= \{\sin(\theta), -\cos(\theta), 0, \cos(\varphi_1), -\cos(\varphi_1) - \sin(\varphi_1)\sin(\varphi_2)\}\\
x4 &= \{0,0,0,1,-1-\cos(\varphi_2)\}\\
x5 &= \{\cos(\theta), \sin(\theta), 0, -\sin(\varphi_1), \sin(\varphi_1) - \cos(\varphi_1)\sin(\varphi_2)\}\\
x6 &= \{0,0,0,\cos(\varphi_1), -2\cos(\varphi_1) - \cos(\varphi_1)\cos(\varphi_2)\}\\
x7 &= \{0,0,0,0,0\}\\
x8 &= \{-\sin(O), \cos(\theta), 0,\ -\cos(\varphi_1), \cos(\varphi_1) + \sin(\varphi_1)\sin(\varphi_2)\}
\end{aligned}
$$

47

$$\mathbf{x9} = \{0,0,0,\ 1,\ -2-\ 2\cos(\varphi_1)^2\cos(\varphi_2)-\cos(\varphi_2)\sin(\varphi_1)^2\}$$

$$x10 = \{0,0,0,-\sin(\varphi_1),2\sin(\varphi_1)\ +\ \cos(\varphi_2)\sin(\varphi_1)\}$$

$$x11 = \{0,0,0,0,0\}$$

$$x12 = \{-\cos(\theta),-\sin(\theta),0,\sin(\varphi_1),-\sin(\varphi_1)+\cos(\varphi_1)\sin(\varphi_2)\}$$

$$x13 = \{0,0,0,\sin(\varphi_1),-2\sin(\varphi_1)\ -\ \cos(\varphi_2)\sin(\varphi_1)\}$$

$$x14 = \{0,0,0,-1,1\ \mathbf{t}\ \ \cos(\varphi_2)\}$$

$$\det(\mathbf{x1}, x2, x3, \mathbf{x4}, x6) = -\cos(\varphi_1)$$

$$\det(\mathbf{x1}, x2, x3, x4, x9) = -1-\cos(\varphi_1)^2\cos(\varphi_2)$$

9 Appendix 3 : Proof of the Lemma in Section 5.

Note X[x] the x-coordinate of the vector field X.

Lemma 1 : The general form of X_i, Y_i and Z_i is :

$$X_i[x] = \sin(\theta - \sum_{j=1}^{i-1}\varphi_j)$$

$$X_i[y] = -\cos(\theta - \sum_{j=1}^{i-1}\varphi_j)$$

$$X_i[\theta] = 0$$

$$X_i[\varphi_0] = 0$$

$$\vdots$$

$$X_i[\varphi_{i-1}] = 0$$

$$X_i[\varphi_i] = \cos\varphi_i$$

$$X_i[\varphi_{i+1}] = -\cos\varphi_i - \sin\varphi_i\sin\varphi_{i+1}$$

$$X_i[\varphi_{i+2}] = \sin\varphi_i[\sin\varphi_{i+1} - \cos\varphi_{i+1}\sin\varphi_{i+2}]$$

$$\vdots$$

$$X_i[\varphi_k] = [\sin\varphi_{k-1} - \cos\varphi_{k-1}\sin\varphi_k]\sin\varphi_i\prod_{j=i+1}^{k-2}\cos\varphi_j$$

$$Y_i[x] = \cos(\theta - \sum_{j=1}^{i}\varphi_j) \qquad\qquad Z_i[x] = -\sin(\theta - \sum_{j=1}^{i}\varphi_j)$$

$$Y_i[y] = \sin(\theta - \sum_{j=1}^{i}\varphi_j) \qquad\qquad Z_i[y] = \cos(\theta - \sum_{j=1}^{i}\varphi_j)$$

$$Y_i[\theta] = 0 \qquad\qquad Z_i[\theta] = 0$$

$$Y_i[\varphi_0] = 0 \qquad\qquad Z_i[\varphi_0] = 0$$

$$Y_i[\varphi_{i-1}] = 0 \qquad\qquad Z_i[\varphi_{i-1}] = 0$$

$$Y_i[\varphi_i] = 0 \qquad\qquad Z_i[\varphi_i] = -1$$

$$Y_i[\varphi_{i+1}] = -\sin\varphi_{i+1} \qquad\qquad Z_i[\varphi_{i+1}] = 1$$

$$Y_i[\varphi_{i+2}] = [\sin\varphi_{i+1} - \cos\varphi_{i+1}\sin\varphi_{i+2}] \qquad\qquad Z_i[\varphi_{i+2}] = 0$$

$$\vdots \qquad\qquad\qquad\qquad\qquad \vdots$$

$$Y_i[\varphi_k] = [\sin\varphi_{k-1} - \cos\varphi_{k-1}\sin\varphi_k]\prod_{j=i+1}^{k-2}\cos\varphi_j \qquad Z_i[\varphi_k] = 0$$

$$\vdots$$

Proof : As in the special case $n = 2$ developed in Section 4.6, we can verify that the lemma holds for $i = 1$ (to do this, simply add $n - 2$ trailers and check the coordinates from φ_1 to φ_n).

Assume now that the lemma holds for X_{i-1}, Y_{i-1} and Z_{i-1}. We will just verify the x, φ_i and φ_k coordinates of $X_i = [Y_{i-1}, Z_{i-1}]$, other computations being similar.

First, Z_{i-1} yields only four non-null coordinates, of which two are constant :

$$Z_{i-1}[x] = \sin(\theta - \sum_{j=1}^{i-1} \varphi_j) \qquad Z_{i-1}[\varphi_{i-1}] = -1$$

$$Z_{i-1}[y] = \cos(\theta - \sum_{j=1}^{i-1} \varphi_j) \qquad Z_{i-1}[\varphi_i] = 1$$

Moreover all the partial derivatives with respect to x and y vanish. Therefore :

$$
\begin{aligned}
X_i[x] &= \sum_{h=1}^{n} \left(Y_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Z_{i-1}[x] - Z_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Y_{i-1}[x] \right) \\
&= -Z_{i-1}[\varphi_{i-1}]\frac{\partial}{\partial\varphi_{i-1}}Y_{i-1}[x] \\
&= \frac{\partial}{\partial\varphi_{i-1}}\cos(\theta - \sum_{j=1}^{i-1}\varphi_j) \\
&= \sin(\theta - \sum_{j=1}^{i-1}\varphi_j)
\end{aligned}
$$

$$
\begin{aligned}
X_i[\varphi_i] &= \sum_{h=1}^{n} \left(Y_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Z_{i-1}[\varphi_i] - Z_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Y_{i-1}[\varphi_i] \right) \\
&= -Z_{i-1}[\varphi_i]\frac{\partial}{\partial\varphi_i}Y_{i-1}[\varphi_i] \\
&= -\frac{\partial}{\partial\varphi_i}(\ -\sin\varphi_i) \\
&= \cos\varphi_i
\end{aligned}
$$

$$
\begin{aligned}
X_i[\varphi_k] &= \sum_{h=1}^{n} \left(Y_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Z_{i-1}[\varphi_k] - Z_{i-1}[\varphi_h]\frac{\partial}{\partial\varphi_h}Y_{i-1}[\varphi_k] \right) \\
&= -Z_{i-1}[\varphi_k]\frac{\partial}{\partial\varphi_i}Y_{i-1}[\varphi_k] \\
&= -\frac{\partial}{\partial\varphi_i}(\prod_{j=i}^{k-2}\cos\varphi_j)[\sin\varphi_{k-1} - \cos\varphi_{k-1}\,sin\,\varphi_k] \\
&= \sin\varphi_i(\prod_{j=i+1}^{k-2}\cos\varphi_j)[\sin\varphi_{k-1} - \cos\varphi_{k-1}\,Sin\,\varphi_k]
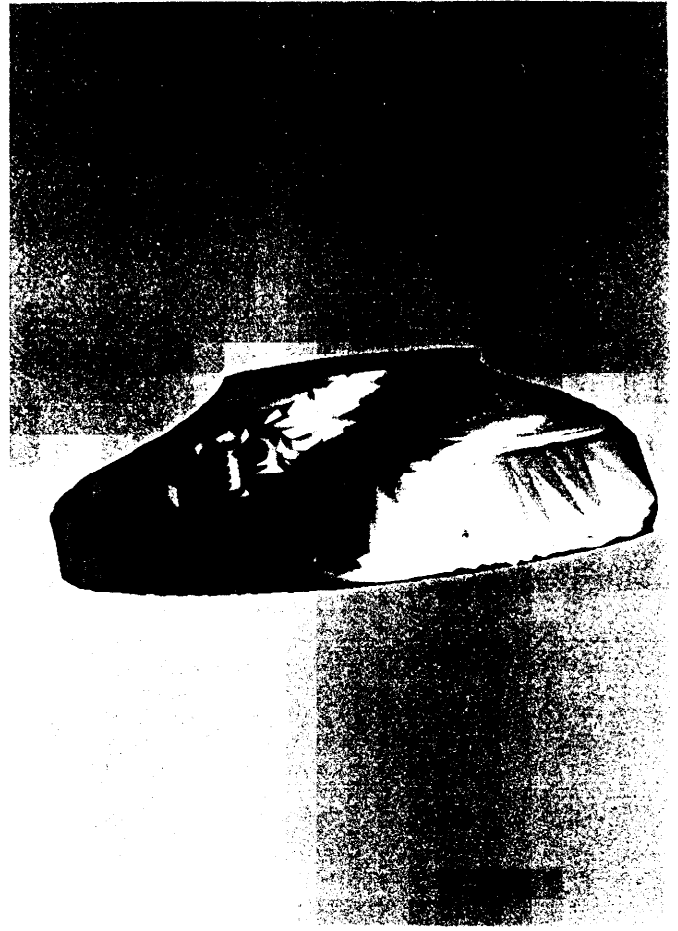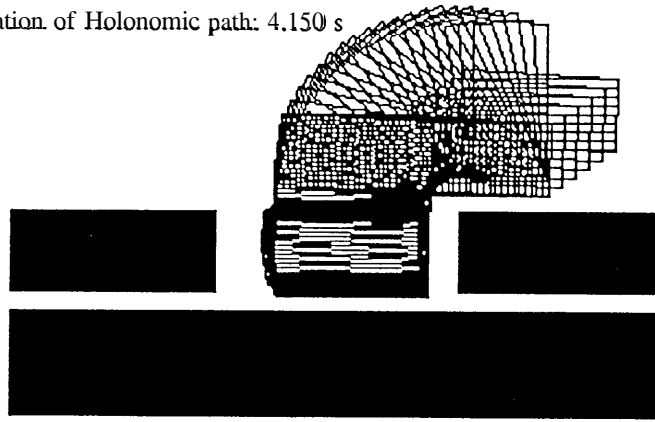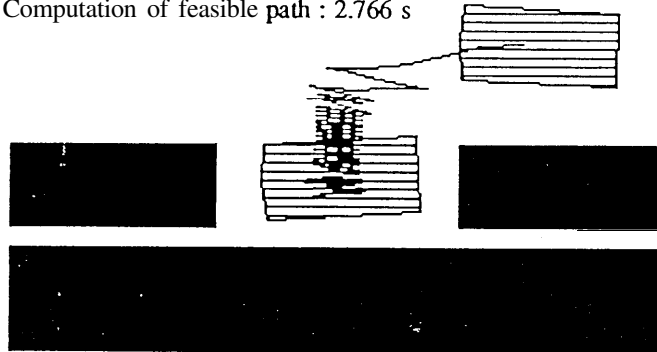\end{aligned}
$$

Q.E.D.

50

Figure 3: A same ball under three different points of view.

Computation of Holonomic path: 4.150 s
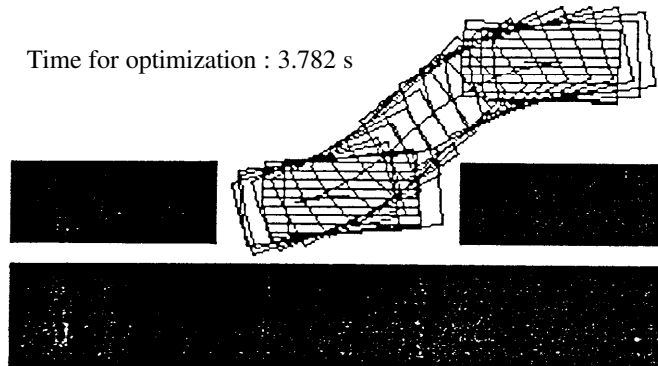
Computation of feasible path : 2.766 s

Time for optimization : 3.782 s

Figure 4: The three steps of the algorithm.
52