TRAJECTORY GENERATION FOR THE N-TRAILER PROBLEM USING GOURSAT NORMAL FORM

by

D. Tilbury, R. Murray, and S. Sastry

Memorandum No. UCB/ERL M93/12

15 February 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TRAJECTORY GENERATION FOR THE N-TRAILER PROBLEM USING GOURSAT NORMAL FORM

by

D. Tilbury, R. Murray, and S. Sastry

Memorandum No. UCB/ERL M93/12

15 February 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TRAJECTORY GENERATION FOR THE N-TRAILER PROBLEM USING GOURSAT NORMAL FORM

by

D. Tilbury, R. Murray, and S. Sastry

Memorandum No. UCB/ERL M93/12

15 February 1993

Trajectory Generation for the N-trailer problem using Goursat Normal Form*

D. Tilbury[†], R. Murray[‡], S. Sastry

Electronics Research Laboratory
Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720

February 15, 1993

Abstract. In this paper, we develop the machinery of exterior differential forms, more particularly the Goursat normal form for a Pfaffian system, for solving nonholonomic motion planning problems, *i.e.* planning problems with non-integrable velocity constraints. We apply this technique to solving the problem of steering a mobile robot with n trailers. We present an algorithm for finding a family of transformations which will display the given system of rolling constraints on the wheels of the robot with n trailers in the Goursat canonical form. Two of these transformations are studied in detail. The Goursat normal form for exterior differential systems is dual to the so-called chained form for vector fields tbat we have studied in our earlier work. Consequently, we are able to give the state feedback law and change of coordinates to convert the N-trailer system into chained form. Three methods for steering chained form systems using sinusoids, piecewise constants and polynomials as inputs are presented.

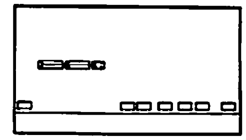
The motion planning strategy is therefore to first convert the N-trailer system into chained form, steer the corresponding chained form system, then transform the resulting trajectory back into the original coordinates. Simulations and frames of movie animations of the N-trailer system for parallel parking and backing into a loading dock using this strategy are also included.

*The research of D. Tilbury and S. Sastry was supported in part by NSF under grant IRI-9014490. The work of R. Murray was supported in part by a grant from the Powell Foundation. D. Tilbury would also like to acknowledge an AT&T Ph.D. Fellowship for partial support of this work.

[†]dawnt@robotics.berkeley.edu

[‡]Address: Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125

1. Introduction



In the past few years there has been a great deal of interest in the generation of motion planning algorithms for robots with nonholonomic or non-integrable velocity constraints in cluttered environments. These constraints on the instantaneous velocities that can be achieved arise from the kinematics of the drive mechanisms of the carts. This work has been a departure from the traditional robot motion planning (see for example [6, 15, 18]) which concentrated on understanding the complexity of the computational effort associated with planning trajectories for robots (with no constraints on their instantaneous velocities) which would avoid both fixed and moving obstacles. Unfortunately the motion plans arising from these more traditional methods often required sideways motion of robot carts with wheels, and as was pointed out by Laumond, most mobile robots are not on castors [19, 20].

In this paper, we consider and solve the motion planning problem for a system consisting of a car-like mobile robot pulling n trailers. This system has been an important canonical example for the work on nonholonomic motion planning ever since it was posed in [22, 28]. The nonholonomic constraints for this system arise from constraining each pair of wheels to roll without slipping. Strictly speaking, if an axle has a differential that keeps the pair of wheels rolling without slipping, then each wheel turns a different amount in accordance with a simple geometric relationship called the Alexander-Maddocks condition [1]. In our system we will neglect this and model the wheels on an axle as being parallel.

The system of a car with n trailers has been viewed as a canonical example because each trailer adds one dimension to the state space of the system (representing its angle with respect to the inertial frame) and one nonholonomic constraint. Regardless of the number of trailers attached, the general system always has two degrees of freedom, corresponding to the driving and steering directions of the front car. It has been shown that every point in the state space is reachable, *i.e.* that the system is completely controllable [22]. The question that is answered in this paper is one of constructive controllability; explicit open loop controls for steering the car with n trailers from an initial to a final position are given.

We first give a brief description of some of the previous work on this problem. A more detailed review of the general nonholonomic motion planning problem can be found in [29] or in a recent collection of papers [24]. A more detailed description of the N-trailer problem and its variations can be found in [21].

Barraquand and Latombe [2] proposed a planner for cars with trailers in a cluttered environment, with an attempt at finding one with a minimal number of backups. The main drawback to their approach was that it required a discretization of the state space followed by an exhaustive search of all possible directions the robot could go at each point. Consequently, the method became computationally infeasible for a large number of trailers. The method, however, worked well in a very cluttered environment since the presence of many obstacles drastically reduced the number of search directions. Related to this work is that of Divelbiss and Wen [8] which uses gradient descent in a discretized input space. Obstacles can be incorporated into their method using potential fields. The convergence properties of their methods are currently under investigation.

Reeds and Shepp [30] proved an interesting result on the minimum length feasible paths for a robot of the Hilare type with bounded turning radius. They showed that the optimal length path belonged to a family of paths that consisted of segments of straight lines and arcs of circles. It seems doubtful that such a method could be generalized easily to a car with n trailers.

A paper by Murray and Sastry [29] studied motion planning for nonholonomic systems, and focused attention on a specific class of systems in so-called "chained form":

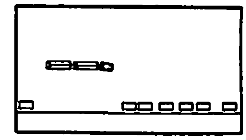
$$\begin{aligned}
\dot{x}_1 &= u_1 \\
\dot{x}_2 &= u_2 \\
\dot{x}_3 &= x_2 u_1 \\
&\vdots \\
\dot{x}_n &= x_{n-1} u_1.
\end{aligned}$$

This class of systems was inspired by some early work of Brockett [3] on optimal control of "canonical systems". In [29], we gave sufficient conditions for converting systems into chained form, and an algorithm (using sinusoids at integrally related frequencies) for steering chained form systems. The theory was used to transform the front-wheel drive car, a car with one trailer, and a hopping robot into chained form, and to find feasible trajectories for these systems using the sinusoidal steering algorithm. However, the car with two trailers did not fit the sufficient conditions and was left an open problem. Recently Sørdalen [32] showed that the system of the car with n trailers could be put into chained form using the coordinates of the n^{th} trailer (rather than those of the cab) for parameterizing the configuration space of the system.

Fernandes, Gurvits and Li [9] used numerical methods for solving constrained optimal control problems associated with nonholonomic motion planning problems, using a perturbation (of the cost functional) to make the singular optimal control problem regular. In other work [10], they also suggested the use of input sinusoids (as basis functions) in a Ritz approximation algorithm for steering nonholonomic systems.

Sinusoids were also used in a method proposed by Sussmann and Liu [33], see also Gurvits and Li [14]. Their method was completely general in that it applied to any controllable nonholonomic system, and used asymptotically high frequency, high magnitude sinusoids to achieve convergence. This method was applied to the system of Hilare with two trailers in [35], but the paths generated were highly oscillatory and impractical, owing to their use of high magnitude and frequency sinusoids.

Divelbiss and Wen [8] have explored a computational approach to the N-trailer problem, in which they discretize the system and use gradient descent in the (discretized) input space to generate a feasible path. Convergence properties of their



method are currently under investigation, but the algorithm has demonstrated good performance in simulation. They are also able to incorporate obstables into the problem formulation, using potential field methods.

Several other methods have been proposed which used piecewise constant inputs [16, 17, 26]. All of these worked best in the cases where the control Lie algebra is nilpotent. Their extension to systems whose control Lie algebra is not nilpotent is not fully satisfactory, requiring a large number of steps to come close to the goal point.

Our current paper is some what different in style from most of the previous work. Instead of focusing on the directions in which the system is allowed to move, namely the two vector fields which correspond to the two degrees of freedom of the system, we have defined the system from the constraints on its velocity. That is, instead of looking at the control system

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2$$

and the distribution spanned by the input vector fields $\Delta = \{g_1, g_2\}$, we consider the exterior differential system orthogonal to this distribution, namely $I = \Delta^{\perp} = \{\alpha^1, \ldots, \alpha^{n-2}\}$. In the context of motion planning, this is in some sense a very natural framework since each α^i is a one-form defined on the tangent space to the configuration space, and represents the constraint that the wheels on the i^{th} axle must roll without slipping.

This system I is called a *Pfaffian system* (of codimension 2); such exterior differential systems and their properties were first studied by Pfaff in the early 1800's. There exists a large body of work on Pfaff's problem in the literature (see [4] for a historical overview). The formulation of the N-trailer problem as an exterior differential system allows us to draw on classical results by Goursat, Engel, Cartan, and others on classification and canonical forms. Most of the relevant results in this area are presented in abbreviated fashion in [4] and are reviewed in Section 2 of this paper. The normal form for Pfaffian systems of codimension 2 that was proposed by Goursat is in fact the dual of chained canonical form as defined above. As in the work of Sørdalen, the calculations for the Goursat normal form are simplified quite considerably by using the coordinates of the last trailer instead of those of the cab to parameterize the configuration space of the multi-trailer system.

After the crash course on exterior differential systems in Section 2, we examine the Pfaffian system associated with a mobile robot towing n trailers. We show in Section 3 that this system can be converted into Goursat's normal form or equivalently chained form. Section 4 is devoted to presenting methods for steering systems in chained form. Three different methods are presented using as inputs sinusoids, piecewise constant inputs (as in [26]) and polynomials. Finally, we apply some of these steering methods to the N-trailers example, and display the results in Section 5. There are movie animations of two of the trajectories; a two-trailer system can be seen parallel-parking by viewing the upper right-hand corner of the odd numbered pages (from the front of the paper to the back), and this same system backing into a loading dock can be seen in the upper left-hand corner of the even numbered pages (from the back of the paper to the front).

2. A Crash Course on Exterior Differential Systems

In this section we give a review of the tools available from the study of exterior differential systems and show how to apply these tools to the problem of finding a feedback transformation which converts a system into chained form. We present only a very brief review of the necessary tools here, concentrating on the computations that must be performed. A much more detailed description can be found in the monograph by Bryant *et al.* [4].

2.1. Exterior algebra. Let V be a vector space over $\mathbb{R}$, which we also refer to using the notation Λ^1. We define a new vector space Λ^2 by defining the wedge product as a skew-symmetric bilinear map which satisfies:

$$\begin{aligned}(a_1\alpha_1 + a_2\alpha_2)\wedge\beta &= a_1(\alpha_1\wedge\beta) + a_2(\alpha_2\wedge\beta)\\ \alpha\wedge(b_1\beta_1 + b_2\beta_2) &= b_1(\alpha\wedge\beta_1) + b_2(\alpha\wedge\beta_2)\\ \alpha\wedge\alpha &= 0\\ \alpha\wedge\beta &= -\beta\wedge\alpha.\end{aligned} \tag{1}$$

That is, $\wedge$ is a bilinear, associative, distributive, non-commutative product mapping $\Lambda^1\times\Lambda^1\to\Lambda^2$. If $\{\sigma_i\}$ is a basis for Λ^1, then $\sigma_i\wedge\sigma_j$, $1\le i<j\le n$ is a basis for Λ^2. It follows that the dimension of of Λ^2 is $\binom{n}{2}$. An element of Λ^2 is called a two-vector.

In a similar way, we define a p-vector $\theta\in\Lambda^p$ by taking the wedge product between p one-vectors and using the rules

$$\begin{gathered}(a\alpha + b\beta)\wedge\alpha_2\wedge\cdots\wedge\alpha_p = a\alpha\wedge\alpha_2\wedge\cdots\wedge\alpha_p + b\beta\wedge\alpha_2\wedge\cdots\wedge\alpha_p\\ \alpha_1\wedge\cdots\wedge\alpha_p = 0 \quad \text{if any} \quad \alpha_i = \alpha_j, i\neq j\\ \alpha_1\wedge\cdots\wedge\alpha_p \text{ changes sign if any two } \alpha_i \text{ are interchanged.}\end{gathered}$$

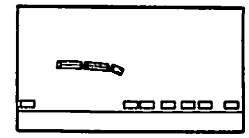
Λ^p consists of all p^{th} order exterior products and has a basis given by $\{\sigma_{h_1}\wedge\cdots\wedge\sigma_{h_p}\}$ where $\{\sigma_i\}$ is a basis for Λ^1 and the h_i's are ordered. Λ^p is a vector space of dimension $\binom{n}{p}$. In particular, $\dim\Lambda^n = 1$ and $\dim\Lambda^k = 0$ if $k>n$. For completeness, we define the set of zero-vectors as $\Lambda^0 = \mathbb{R}$.

The wedge product is a very powerful tool which can be used to great advantage in calculations. We will make frequent use of the following facts:

Proposition 1. *The vectors $v_1,\ldots,v_p\in\Lambda^1$ are linearly dependent if and only if $v_1\wedge\cdots\wedge v_p = 0$.*

Corollary 1.1. *Let $\xi, v_i\in\Lambda^1$. If $\xi\wedge x_1\wedge\cdots\wedge v_p = 0$ then*

$$\xi = \sum_{i=1}^{p}\alpha_i v_i \qquad \alpha_i\in\mathbb{R}.$$



Just as in the case of polynomials, it is often desirable to speak of a vector of mixed order (or unknown order). Using the wedge product, one can define an algebra over the set of exterior forms. Let $\Lambda = \Lambda^0 \oplus \Lambda^1 \oplus \cdots \oplus \Lambda^n$ and define multiplication between two elements of Λ using the wedge product. The wedge product is a bilinear, associative, distributive, skew product which maps $\Lambda^r \times \Lambda^s \to \Lambda^{r+s}$ and hence $\Lambda \times \Lambda \to \Lambda$. We say an element $\xi \in \Lambda$ is homogeneous of order p if $\xi \in \Lambda^p$; *i.e.* it is a p-vector. A exterior form is non-homogeneous if it has components of different orders.

If V is a vector space of dimension n, its dual, V^*, is also a vector space of dimension n. The exterior product over V^* can be used to form the vector space $\Omega^p(V) := \Lambda^p(V^*)$. An element $\alpha \in \Omega^p$ is called a p-form.

2.2. Differential forms. Given a manifold M of dimension n, the tangent space of M at a point x is a vector space of dimension n, denoted T_xM. The vector space $\Lambda^p(T_xM)$ consists of all p-vectors constructed from tangent vectors in T_xM. By attaching the vector space $\Lambda^p(T_xM)$ to each point $x \in M$, we get a bundle structure on M, which we write as $\Lambda^p(M)$. Similarly, the bundle $\Omega^p(M)$ is defined by using the dual space T_x^*M. We call a element $\omega \in \Omega^p(M)$ an exterior differential p-form on M.

Relative to a local coordinate chart, we describe the tangent and cotangent bundles by choosing a local basis:

$$\begin{aligned} T_xM &= \mathrm{span}\{\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\} \\ T_x^*M &= \mathrm{span}\{dx_1, \ldots, dx_n\}, \end{aligned}$$

where

$$dx_i(\frac{\partial}{\partial x_j}) = \delta_{ij}.$$

A p-form ω on M can be represented in this basis as

$$\omega(x) = \sum_{i_1 < \cdots < i_p} \omega_{i_1 \cdots i_p}(x) dx_{i_1} \wedge \cdots \wedge dx_{i_p}.$$

We say that ω is smooth if the coefficient functions $\omega_{i_1 \cdots i_p}$ are smooth functions of x for any choice of coordinate chart.

Let $\Omega(M)$ be the algebra of exterior differential forms on M. The *exterior derivative* on $\Omega(M)$ is the unique map $d : \Omega^r \to \Omega^{r+1}$ which satisfies the following properties:

(1) If $f \in \Omega^0(M) = C^\infty(M)$ then $df = \sum \frac{\partial f}{\partial x_i} dx_i$(relative to a local coordinate chart).
(2) If $\theta \in \Omega^r$, $\sigma \in \Omega^s$ then $d(\theta \wedge \sigma) = d\theta \wedge \sigma + (-1)^r \theta \wedge d\sigma$.
(3) $d^2 = 0$.

These rules are extremely useful in computations.

We will make frequent use of the following lemma, which relates the exterior derivative of a one-form to the Lie bracket between two vector fields.

Lemma 2. *Let $\omega \in \Omega^1(M)$ and let X and Y be smooth vector fields on M. Then*

$$dw(X,Y) = X\omega(Y) - Y\omega(X) - \omega([X,Y]).$$

Proof. It suffices to show that the lemma is true for a basis element, and hence for $\omega = f dg$. On the one hand, we have

$$\begin{aligned} d\omega(X,Y) &= (df \wedge dg)(X,Y) \\ &= df(X)dg(Y) - df(Y)dg(X) \\ &= X(f)Y(g) - Y(f)X(g). \end{aligned}$$

Furthermore,

$$\begin{aligned} &X\omega(Y) - Y\omega(X) - \omega([X,Y]) \\ &\qquad = X(fY(g)) - Y(fX(g)) - f(XY(g) - YX(g)) \\ &\qquad = X(f)Y(g) - Y(f)X(g), \end{aligned}$$

and the lemma is proved. □

This lemma gives the following version of Frobenius's theorem.

Theorem 3 (Frobenius). *Let Δ be a C^∞ distribution of dimension k on M, an n-dimensional manifold. Δ is involutive if and only if there exist $n-k$ linearly independent one-forms $\omega^{k+1}, \ldots, \omega^n$ which vanish on Δ and satisfy*

$$d\omega^i = \sum_{j=k+1}^{n} \theta^i_j \wedge \omega^j \qquad i = k+1, \ldots, n \tag{2}$$

for some set of one-forms θ^r_j.

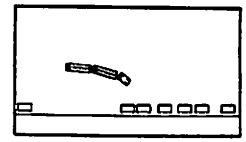
Proof. The proof follows from application of Lemma 2 and Frobenius' theorem for vector fields. Let X, Y be two vector fields in Δ. Then

$$[X,Y] \in \Delta \iff \omega^i([X,Y]) = 0 \qquad i = k+1, \ldots, n$$

since the ω's annihilate Δ. Now applying Lemma 2 we have

$$\begin{aligned} [X,Y] \in \Delta &\iff -d\omega^i(X,Y) + X\omega^i(Y) - Y\omega^i(X) = 0 \\ &\iff d\omega^i(X,Y) = 0 \qquad i = k+1, \ldots, n. \end{aligned}$$

It follows that $d\omega^i$ must have the form in equation (2) since $d\omega$ is annihilated on all vectors $X, Y \in \Delta$ and $\{\omega^{k+1}, \ldots, \omega^n\}$ form a basis for the space of one-forms which annihilate Δ. □



2.3. Pfaffian Exterior Differential Systems. Formally, an *exterior differential system* is given by an ideal $\mathcal{I} \subset \Omega(M)$ that is closed under exterior differentiation. Recall that an ideal $\mathcal{I}$ satisfies

$$\alpha \in \mathcal{I}, \beta \in \Omega(M) \quad \Longrightarrow \quad \alpha \wedge \beta \in \mathcal{I}.$$

We will be primarily interested in the special case of exterior differential systems which are generated by a set of nonholonomic constraints and we focus on that case here.

A *Pfaffian system* is an exterior differential system which is generated by a set of linearly independent one-forms. Let I be a codistribution spanned by a set of linear independent one-forms $\{\alpha^i\}$, $i = 1, \ldots, s$. The ideal generated by I is

$$\mathcal{I} = \{I\} = \{\sigma \in \Omega : \sigma \wedge \alpha^1 \cdots \wedge \alpha^s = 0\}.$$

For an ideal generated by a set of one-forms, each element in the ideal has the form

$$\xi = \sum_{j=1}^{s} a_{ij} \theta^j \wedge \alpha^j$$

for some $\theta^j \in \Omega$.

It is also possible to rephrase Frobenius's Theorem in a concise way using ideals. Let $\mathcal{I}$ be the ideal generated by $\{\alpha^1, \ldots, \alpha^s\}$ and write $d\mathcal{I}$ for the set consisting of the exterior derivative of all elements of $\mathcal{I}$. We say that $\mathcal{I}$ is *integrable* if there exist functions $h_1, \ldots, h_s$ such that $\mathcal{I} = \{dh_1, \ldots, dh_s\}$. The Frobenius theorem asserts that the following set of relationships hold:

$$\begin{aligned}
\mathcal{I} \text{ is integrable } &\Longleftrightarrow d\mathcal{I} \subset \mathcal{I} \\
&\Longleftrightarrow d\alpha^i \wedge \alpha^1 \wedge \cdots \wedge \alpha^s = 0 \\
&\Longleftrightarrow d\alpha^i = \sum_{j=1}^{s} \theta^i_j \wedge \alpha^j \quad \text{for some } \theta^i_j,\ i = 1, \ldots, s \\
&\Longleftrightarrow d\alpha^i \equiv 0 \mod \mathcal{I}.
\end{aligned} \tag{3}$$

The last relationship in equation (3) uses the notion of *congruence*. Given two forms $\omega, \xi \in \Omega$, we write $\omega \equiv \xi \mod \mathcal{I}$ if there exists an exterior form $\eta \in \mathcal{I}$ such that $\omega = \xi + \eta$. If I is a set of one-forms (and hence not an ideal) then we write $\omega \equiv \xi \mod I$ if there exist exterior forms $\alpha \in I$ and $\eta \in \Omega$ such that $\omega = \xi + \eta \wedge \alpha$. It follows that if I is the generator set for an ideal $\mathcal{I}$, then $\omega \mod \mathcal{I} = \omega \mod I$. In the case that $\mathcal{I}$ is generated by one-forms $\{\alpha_i\}$, we will often make use of the relationship

$$\omega \mod \mathcal{I} \equiv 0 \iff \omega = \sum \theta_i \wedge \alpha^i \quad \text{for some } \theta_i \in \Omega.$$

Although conceptually simple, mod-ing out by a set of one-forms can sometimes require considerable effort. For example, given the expression

$$dx_1 \wedge dx_2 + dx_2 \wedge dx_3 \mod dx_3,$$

it is easy to see that this is congruent to $dx_1 \wedge dx_2$. One merely sets all components containing a factor of dx_3 to zero. However, for the expression

$$(dx_1 + dx_3) \wedge dx_2 \mod dx_2 + dx_3$$

this simple prescription will not work. The one-form must be rewritten in terms of an appropriate basis. For example, the above expression can be rewritten as

$$(dx_1 + dx_3) \wedge (dx_2 + dx_3) - dx_1 \wedge dx_3 \mod dx_2 + dx_3$$

and this is clearly congruent to $-dx_1 \wedge dx_3$. More generally, to compute ω mod $\alpha^1, \ldots, \alpha^s$, one must rewrite ω in terms of a basis which includes the α^i's as elements.

2.4. The derived flag. Let $I = \operatorname{span}\{\omega^1, \ldots, \omega^s\}$ be a smooth codistribution on M. The exterior derivative induces a mapping $\delta : I \to \Omega^2(M)/I$:

$$\delta : \lambda \mapsto d\lambda \bmod I \in \Omega^2(M).$$

The mapping δ is a linear mapping over $C^\infty(M)$:

$$\begin{aligned} \delta(f\alpha + g\beta) &= df \wedge \alpha + fd\alpha + dg \wedge \beta + gd\beta \mod I \\ &= fd\alpha + gd\beta \mod I \\ &= f\delta(\alpha) + g\delta(\beta). \end{aligned}$$

It follows that the kernel of δ is a codistribution on M (*i.e.* at each point $p \in M$, the kernel of δ is a linear subspace of T_p^*M). We call this subspace $I^{(1)}$, the *first derived system* of I:

$$I^{(1)} = \ker \delta = \{\lambda \in I : d\lambda \bmod I \equiv 0\}.$$

We can represent $I^{(1)}$ using a set of one-forms, but it is important to note that the basis for $I^{(1)}$ may not be a simple subset of the basis for I. Linear combinations of basis elements (over the ring of smooth functions on M) must be searched to find a basis for the derived system.

Since $I^{(1)}$ is itself a Pfaffian system, we can continue this construction and generate a nested sequence of codistributions

$$I = I^{(0)} \supset I^{(1)} \supset \cdots \supset I^{(N)}. \tag{4}$$

If the dimension of each $I^{(i)}$ is constant, then this construction terminates for some finite integer N. In this case, we call equation (4) the *derived flag* of I and N the *derived length.*

The derived flag describes the integrability properties of the ideal generated by I. If I is completely integrable, then by Frobenius's theorem we have $I^{(1)} = I^{(0)}$, *i.e.* the length of the derived flag is zero. In fact, $I^{(N)}$ is always integrable since by definition $dI^{(N)} \bmod I^{(N)} \equiv 0$. $I^{(N)}$ is the *largest* integrable subsystem contained in I. Thus if $I^{(N)}$ is not empty, then there exist functions $h_1, \ldots, h_r$ such that $\{dh_i\} \subset \{I\}$. In the context of control theory, this means that the system is not controllable since there exist algebraic functions which provide a foliation of the

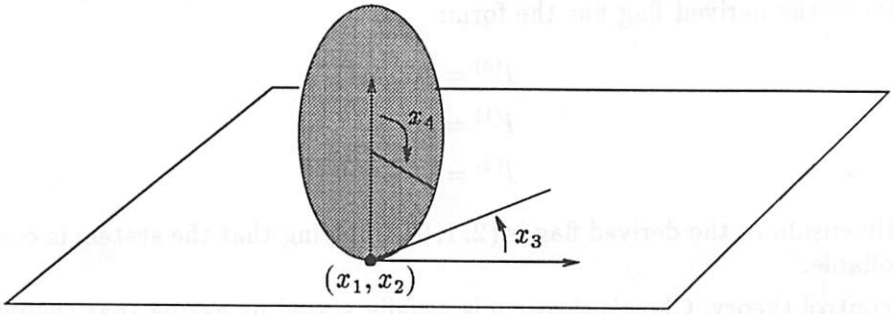


FIGURE 1. Penny rolling on a plane. The state space is defined by the (x_1, y_1) position of the penny, x_3, the angle of the penny with respect to the reference frame, and x_4 the amount the penny is rotated with respect to the vertical.

state space and it is impossible to move from one leaf of the foliation to another. The converse of this controllability result is provided by *Chow's Theorem.*

Theorem 4 (Chow). *Let $I = \{\alpha^1, \ldots, \alpha^s\}$ represent a set of constraints and assumed that the derived flag of the system exists. Then, there exists a path $x(t)$ between any two points satisfying $\alpha^i(x)\dot{x} = 0$ for all i if and only if there exists an N such that $I^{(N)} = \{0\}$.*

Example 1. Consider the kinematic model of a penny rolling on a plane, as shown in Figure 1. Let $x \in \mathbb{R}^4$ denote the configuration of the penny, with (x_1, x_2) being the location of the penny on the plane, x_3 the angle that the penny makes with a fixed line on the plane, and x_4 the angle of a fixed radial line on the penny with respect to the vertical. We take the radius of the penny as 1. The constraints for the penny are that it roll in the direction it is pointing, with no slipping:

$$\alpha^1 = \cos x_3 dx_1 + \sin x_3 dx_2 - dx_4 \tag{5}$$

$$\alpha^2 = \sin x_3 dx_1 - \cos x_3 dx_2. \tag{6}$$

The exterior derivatives of α^1 and α^2 are given by

$$d\alpha^1 = -\sin x_3\, dx_3 \wedge dx_1 + \cos x_3\, dx_3 \wedge dx_2 \tag{7}$$

$$d\alpha^2 = \cos x_3\, dx_3 \wedge dx_1 + \sin x_3\, dx_3 \wedge dx_2. \tag{8}$$

It is easy to check that the following relationships hold

$$d\alpha^1 \wedge \alpha^1 = -dx_1 \wedge dx_2 \wedge dx_3 + \cos x_3 dx_2 \wedge dx_3 \wedge dx_4 - \sin x_3 dx_1 \wedge dx_3 \wedge dx_4$$

$$d\alpha^1 \wedge \alpha^1 \wedge \alpha^2 = 0$$

$$d\alpha^2 \wedge \alpha^1 \wedge \alpha^2 = -dx_1 \wedge dx_2 \wedge dx_3 \wedge dx_4 \neq 0,$$

and hence the derived flag has the form:

$$\begin{aligned} I^{(0)} &= \{\alpha^1, \alpha^2\} \\ I^{(1)} &= \{\alpha^1\} \\ I^{(2)} &= \{0\}. \end{aligned} \tag{9}$$

The dimension of the derived flag is $(2, 1, 0)$, implying that the system is completely controllable.

In control theory, Chow's theorem is usually stated by asking that the involutive closure of the distribution $I^\perp$ span the tangent space at each point $x \in M$. The connection between the Lie algebra formulation of Chow's theorem and the exterior differential system formulation is made with the following lemma.

Lemma 5. *If $I = \Delta^\perp$ then $I^{(1)} = (\Delta + [\Delta, \Delta])^\perp$.*

Proof. Follows from Lemma 2. □

This lemma allows us to compute the derived flag for a system given the distribution $\Delta = I^\perp$. Define the nested set of distributions $E_0 \subset E_1 \subset \cdots \subset E_N$ as

$$\begin{aligned} E_0 &= \Delta \\ E_i &= E_{i-1} + [E_{i-1}, E_{i-1}]. \end{aligned}$$

This sequence terminates if the dimension of each E_i is constant, and it follows from Lemma 5 that $I^{(i)} = E_i^\perp$.

Remark 1. When doing computations with exterior differential systems, it is convenient to choose a basis of one-forms whose structure matches that of the derived flag. We say that a basis $\{\alpha^i\}$ is *adapted to the derived flag* if

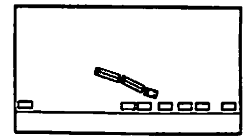
$$I^{(i)} = \{\alpha^1, \ldots, \alpha^{s_i}\},$$

where s_i is a strictly decreasing sequence of integers. In other words, an adapted basis is one in which the derived systems are calculated by dropping elements from the end of the basis. An adapted basis can be calculated by computing the derived flag and then choosing the basis elements starting with a basis for $I^{(N-1)}$ and proceeding backwards.

2.5. Pfaff's problem and Engel's theorem. The simplest type of normal form for a nonholonomic system involves finding a normal form for a single constraint.

Theorem 6 (Pfaff's problem). *Suppose α is a one-form which satisfies $(d\alpha)^{r+1} \wedge \alpha = 0$, $(d\alpha)^r \wedge \alpha \neq 0$. Then there exist coordinates such that*

$$\alpha = dx_1 + x_2 dx_3 + \cdots + x_{2r} dx_{2r+1}.$$



The proof uses a number of tools that are beyond the scope of this paper. In the $r = 1$ case, the proof reduces to proving that there exist two functions f_1 and f_2 which satisfy

$$\begin{aligned} d\alpha \wedge \alpha \wedge df_1 &= 0 \\ \alpha \wedge df_1 \wedge df_2 &= 0 \end{aligned} \quad \text{and} \quad \begin{aligned} \alpha \wedge df_1 &\neq 0 \\ df_1 \wedge df_2 &\neq 0. \end{aligned} \tag{10}$$

Given f_1 and f_2, α can be scaled such that

$$\alpha = df_2 + g df_1 =: dx_1 + x_2 dx_3.$$

The Pfaff theorem guarantees that these equations have a solution (it need not be unique).

In the case of a single constraint in $\mathbb{R}^3$, Pfaff's theorem shows that if the constraint is non-integrable then the corresponding control system can be written in chained form. This follows because if α is not integrable then $d\alpha \wedge \alpha \neq 0$ but $(d\alpha)^2 \wedge \alpha = 0$ by a dimension count. Therefore, we can apply Theorem 6 (with a relabeling of coordinates) to conclude that

$$\alpha = d\xi_3 - \xi_2 d\xi_1.$$

A basis for the right null space of this constraint is then given by

$$g_1 = \frac{\partial}{\partial \xi_1} + \xi_2 \frac{\partial}{\partial \xi_3} \qquad g_2 = \frac{\partial}{\partial \xi_2},$$

which is the chained form for two input vector fields in $\mathbb{R}^3$.

Engel's theorem applies to the case of two non-integrable constraints in $\mathbb{R}^4$.

Theorem 7 (Engel's theorem). *Let I be a two-dimensional codistribution on $\mathbb{R}^4$ with $\dim I^{(1)} = 1$ and $\dim I^{(2)} = 0$. Then there exist local coordinates such that*

$$I = \{d\xi_4 - \xi_3 d\xi_1, d\xi_3 - \xi_2 d\xi_1\}. \tag{11}$$

Proof. Choose a basis $I = \{\alpha^1, \alpha^2\}$ which is adapted to the derived flag. It follows that $d\alpha^1 \wedge \alpha^1 \neq 0$ and $(d\alpha^1)^2 \wedge \alpha^1 = 0$ (by dimension count). Hence we can use Pfaff's theorem to find coordinates such that $\alpha^1 = d\xi_4 - \xi_3 d\xi_1$.

To determine ξ_2, we use the structure of α^2. Since $\alpha^1 \in I^{(1)}$, we have $d\alpha^1 \wedge \alpha^1 \wedge \alpha^2 = 0$. But $d\alpha^1 = -d\xi_3 \wedge d\xi_1$ and hence

$$\alpha^2 \equiv a\, d\xi_3 + b\, d\xi_1 \mod \alpha^1.$$

Since $\alpha^2 \neq 0$, it follows that we can not have both a and $b = 0$. We split the proof into two cases.

Case 1: ($a \neq 0$). Since α^2 is only determined mod α^1, we are free to scale α^2 by any nonzero function. Hence

$$\frac{1}{a}\alpha^2 \equiv d\xi_3 + \frac{b}{a} d\xi_1 \mod \alpha^1 \tag{12}$$

and choosing $\xi_2 = -b/a$ yields a basis for the codistribution which is in Engel's normal form. Notice that the basis is a transformed version of the original basis, namely,

$$\begin{aligned}\bar\alpha_1 &= \alpha_1 &&= d\xi_4 - \xi_3 d\xi_1\\ \bar\alpha_2 &= \frac{1}{a}\alpha_2 + \lambda\alpha_1 &&= d\xi_3 - \xi_2 d\xi_1\end{aligned}$$

where λ is chosen such that equation (12) becomes an equality.

Case 2: $(b \neq 0)$. In this case we can scale α^2 so that

$$\frac{1}{b}\alpha^2 \equiv \frac{a}{b}d\xi_3 + d\xi_1 \mod \alpha_1.$$

Defining $\xi_2 = -a/b$ gives the normal form

$$\begin{aligned}\bar\alpha_1 &= d\xi_4 - \xi_3 d\xi_1\\ \bar\alpha_2 &= d\xi_1 - \xi_2 d\xi_3.\end{aligned}$$

It turns out that this normal form is diffeomorphic to Goursat form via the following change of coordinates:

$$\begin{aligned}\eta_1 &= \xi_3\\ \eta_2 &= -\xi_2\\ \eta_3 &= -\xi_1\\ \eta_4 &= \xi_4 - \xi_1\xi_3\end{aligned} \quad\Longrightarrow\quad \begin{aligned}\alpha^1 &= d\eta_4 - \eta_3 d\eta_1\\ \alpha^2 &= d\eta_3 - \eta_2 d\eta_1.\end{aligned}$$

Hence the transformed basis is in Engel's normal form. □

2.6. Goursat normal form. We now turn to the more general case of $n-2$ constraints on an n-dimensional manifold M. Let I be a codistribution on M whose derived flag satisfies $\dim I^{(i)} = n - i - 2$.

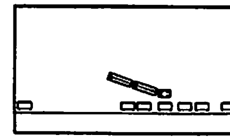
Theorem 8 (Goursat normal form). *Let U be an open subset of $\mathbb{R}^n$ and $I = \{\alpha^1, \ldots, \alpha^s\}$ be a collection of $s = n-2$ smooth, linearly independent one-forms defined on U. If there exists a one-form $\pi \neq 0 \mod I$ such that*

$$\begin{aligned}&d\alpha^i \equiv -\alpha^{i+1}\wedge\pi \mod \alpha^1,\ldots,\alpha^i \qquad i = 1,\ldots,s-1\\ &d\alpha^s \neq 0 \mod I\end{aligned} \tag{13}$$

then there exists a set of coordinates ξ such that

$$I = \{d\xi_n - \xi_{n-1}d\xi_1, \ldots, d\xi_3 - \xi_2 d\xi_1\}.$$

A few comments on the statement of this theorem are in order. The conditions of the theorem require the existence of a special basis $\{\alpha^i\}$ and a special one-form π. A quick calculation shows that the basis $\{\alpha^i\}$ is adapted to the derived flag of the system and hence if we start with an adapted basis, the real requirement is the existence of a one-form π which satisfies the congruences. Determining π can involve a further scaling of the adapted basis which preserves the adapted structure



(see Section 3.2 for an example). For most examples, π can be determined by a combination of physical insight and repeated guessing.

A complete proof of this theorem can be found in [4]. It can be summarized in the following algorithm for converting a system into Goursat form (see [12] for the feedback linearization version of this algorithm, on which this is based).

Algorithm 1. Given a codistribution $I = \{\omega^1, \ldots, \omega^s\}$ with $s = n-2$, the following steps are required:

(1) Construct a basis $I = \{\alpha^1, \ldots, \alpha^s\}$ which is adapted to the derived flag. Check the Goursat congruences to ensure they are satisfied for some π.
(2) It follows from the congruences that α^1 and α^2 satisfy $d\alpha^1 \wedge \alpha^1 \wedge \alpha^2 = 0$ and hence the proof of Engel's theorem can be used to find coordinates such that
$$\alpha^1 = d\xi_n - \xi_{n-1} d\xi_1$$
$$\alpha^2 = d\xi_{n-1} - \xi_{n-2} d\xi_1.$$
This may involve finding a new basis which preserves the adapted structure, as well as a change of coordinates, to convert between the two normal forms in the proof of Engel's theorem.
(3) The remaining coordinates are determined by simple differentiation. Given ξ_i we determine ξ_{i-1} by algebraically solving the equation
$$\alpha^{n-i+1} \equiv d\xi_i + \xi_{i-1} d\xi_1 \mod \alpha^1, \ldots, \alpha^{n-i+1}.$$
The proof of Goursat's theorem is to essentially show that this equation always has a solution.

Example 2. Consider again the rolling penny from the previous example. The basis
$$I = \{\alpha^1, \alpha^2\} \qquad \begin{aligned} \alpha^1 &= \cos x_3 dx_1 + \sin x_3 dx_2 - dx_4 \\ \alpha^2 &= \sin x_3 dx_1 - \cos x_3 dx_2 \end{aligned}$$
was shown to be a basis adapted to the derived flag.

Next, we search for a one-form π which satisfies the Goursat congruences. Since this system is low dimensional, the algebra needed to find π is straightforward. Define α^3 and α^4 so as to complete the basis:
$$\begin{aligned} \alpha^3 &= \cos x_3 dx_1 + \sin x_3 dx_2 \\ \alpha^4 &= dx_3. \end{aligned}$$
π must satisfy
$$d\alpha^1 = -\alpha^2 \wedge \pi \mod \alpha^1; \tag{14}$$
Setting $\pi = \lambda_3 \alpha^3 + \lambda_4 \alpha^4$, equation (14) gives
$$\begin{aligned} \lambda_3 &= 0 \\ \lambda_4 &= 1 \end{aligned} \qquad \Longrightarrow \qquad \pi = dx_3.$$

We now proceed to apply the algorithm for converting the system to Goursat form. In this case, only the Engel's step is required. We begin by solving Pfaff's problem. The first partial differential equation to be solved is $d\alpha^1 \wedge \alpha^1 \wedge df_1 = 0$, which yields

$$(-dx_1 \wedge dx_2 \wedge dx_3 - \sin x_3 dx_1 \wedge dx_3 \wedge dx_4 + \cos x_3 dx_2 \wedge dx_3 \wedge dx_4) \wedge df_1 = 0.$$

This equation has a trivial solution given by $f_1 = x_3$. The second partial differential equation, $\alpha^1 \wedge df_1 \wedge df_2 = 0$, then becomes

$$(\cos x_3 dx_1 \wedge dx_3 + \sin x_3 dx_2 \wedge dx_3 + dx_3 \wedge dx_4) \wedge df_2 = 0,$$

which has a solution given by $f_2 = x_1 \cos x_3 + x_2 \sin x_3 - x_4$.

Finally, we solve for the remaining coordinate by finding functions a and b such that

$$\alpha^1 = a df_1 + b df_2.$$

This is a completely algebraic problem which has a solution given by

$$\begin{aligned} a &= x_1 \sin x_3 - x_2 \cos x_3 \\ b &= 1. \end{aligned}$$

Combining all of these calculations, we define

$$\begin{aligned} \xi_1 &= x_3 \\ \xi_3 &= -x_1 \sin x_3 + x_2 \cos x_3 \\ \xi_4 &= x_1 \cos x_3 + x_2 \sin x_3 - x_4 \end{aligned}$$

which gives $\alpha^1 = d\xi_4 - \xi_3 d\xi_1$.

We now define ξ_2 by examining α^2. From the proof of Engel's theorem we must have

$$\alpha^2 \equiv a d\xi_3 + b d\xi_1 \mod \alpha^1.$$

Performing all calculations in the original coordinates, this relationship becomes

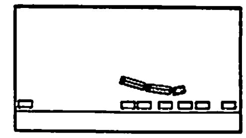
$$\begin{aligned} &\sin x_3 dx_1 - \cos x_3 dx_2 \equiv \\ &\qquad a(-\sin x_3 dx_1 + \cos x_3 dx_2 - (x_1 \cos x_3 + x_2 \sin x_3) dx_3) + b dx_3 \mod \alpha^1 \end{aligned}$$

from which it follows that $a = -1$ and $b = -(x_1 \cos x_3 + x_2 \sin x_3)$. This is case 1 in the proof of Engel's theorem and hence

$$\xi_2 = b/a = (x_1 \cos x_3 + x_2 \sin x_3)$$

and $\bar{\alpha}^2 = -\alpha^2$ is the new basis element. This completes the change of basis and change of coordinates, and the resulting system is in Goursat normal form.

A more complex example, the N-trailer system, is the subject of Section 3.



2.7. Converting systems to chained form. Chained form is dual to the Goursat normal form presented above. That is, a system with constraints in Goursat normal form can always be written as a control system in chained form by choosing

$$\begin{aligned} g_1 &= \frac{\partial}{\partial x_1} + x_2 \frac{\partial}{\partial x_3} + \cdots + x_{n-1} \frac{\partial}{\partial x_n} \\ g_2 &= \frac{\partial}{\partial x_2}, \end{aligned}$$

which forms a basis for the distribution annihilated by I. Thus, we can formulate the problem of finding a basis for the constraints which is in Goursat form as the problem of finding a feedback transformation to convert a system to chained form

The Goursat congruences are somewhat unsatisfying since they require the existence of a one-form π. Necessary and sufficient conditions for the existence of such a π, and hence for converting a set of constraints into Goursat normal form, were presented in [27]. We summarize the main result here.

Let $I = \text{span}\{\omega^1, \ldots, \omega^s\}$ be a codistribution on $\mathbb{R}^n$ and write $\Delta = I^\perp$ for the distribution which spans the null space of the codistribution. We define two nested sets of distributions:

$$\begin{aligned} E_0 &= \Delta & F_0 &= \Delta \\ E_1 &= E_0 + [E_0, E_0] & F_1 &= F_0 + [F_0, F_0] \\ E_2 &= E_1 + [E_1, E_1] & F_2 &= F_1 + [F_1, F_0] \\ &\vdots & &\vdots \\ E_{i+1} &= E_i + [E_i, E_i] & F_{i+1} &= F_i + [F_i, F_0]. \end{aligned} \tag{15}$$

Under the assumption that each distribution is constant rank, the two sequences have finite length (possibly different).

The filtration $\{F_i\}$ is the the one which usually appears in the context of nonlinear controllability and feedback linearization. In particular, F_i consists of all brackets up to order i. The distribution E_i also contains all brackets of order i, but may contain additional Lie products of higher order. This is due to the recursive construction of E_i, as opposed to the iterative construction of F_i. The filtration E_i is precisely the sequence of distributions which is perpendicular to the derived flag of $I = \Delta^\perp$.

Theorem 9 ([27]). *Given a 2-dimensional distribution $\Delta = I^\perp$ such that*

$$\dim E_i = \dim F_i = i + 2 \qquad i = 0, \ldots, n-2,$$

there exists a basis $\{\alpha^1, \ldots, \alpha^s\}$ for I which is in Goursat normal form.

This theorem allows us to completely characterize the set of systems which are equivalent to a system in chained (or Goursat) form in the case that the relative growth vector of the system is $\sigma = (2, 1, \ldots, 1)$. It can be shown that the N-trailer problem satisfies the conditions of Theorem 9 and hence it can be converted to chained form.

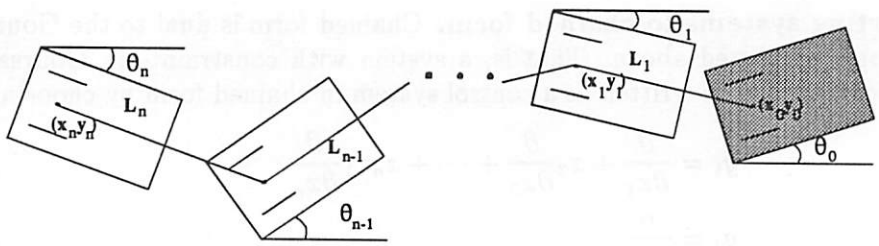


FIGURE 2. The mobile robot Hilare with n trailers.

3. The Motion Planning Problem for the N-trailer system

In this section, we define the Pfaffian system (set of one-forms which represent the velocity constraints) for the N-trailer problem and calculate its derived flag. We then show how the system can be converted into either Goursat normal form (following Theorem 8 and Algorithm 1) or its dual, chained form. Although the calculations in this section assume a particular configuration of the mobile robot and trailer system, we will show that our model is general enough to encompass not only the specific choice we have made but also a front-wheel drive car pulling trailers and the luggage trains found in airports.

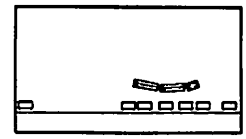
3.1. The system of rolling constraints and its derived flag. Consider a mobile robot such as Hilare[1] with n trailers attached, as in Figure 2. Each trailer is attached to the body in front of it by a rigid bar, and the rear set of wheels of each body is constrained to roll without slipping. The trailers are assumed to be identical, but to have possibly different link lengths L_i. The x, y coordinates of a midpoint between the two wheels are referred to as (x_i, y_i) and the hitch angles (all measured with respect to the horizontal) are θ_i. The connections between the bodies give rise to the following constraints:

$$\begin{aligned} x_i &= x_{i-1} - L_i \cos\theta_i \\ y_i &= y_{i-1} - L_i \sin\theta_i, \end{aligned} \tag{16}$$

$i = 1, 2, \ldots, n$ for the general case with n trailers. These constraints are holonomic and will reduce the dimension of the configuration space, since the positions (x_i, y_i) for $i \geq 1$ can be expressed in terms of $x_0, y_0, \theta_0, \ldots, \theta_i$. By symmetry, (x_i, y_i) for $i < n$ can also be expressed in terms of $x_n, y_n, \theta_n, \theta_{n-1}, \ldots, \theta_i$. For our purposes it will be far more useful to use as configuration space variables the x, y coordinates of a point on the n^{th} trailer and the $n+1$ hitch angles: $x_n, y_n, \theta_n, \ldots, \theta_0$ because the calculations that follow are vastly simplified.[2] We will refer to the state space as $x = (x_n, y_n, \theta_n, \ldots, \theta_1, \theta_0)$. We have assumed that the bodies are connected between

[1]The Hilare family of mobile robots resides at LAAS in Toulouse, see for example [7, 13].

[2]The intuition for this comes from the oft repeated dictum: "when backing up a car with a trailer, keep your eye on the hind part of the trailer". Of course, the generalization to this dictum is: when driving a car with n trailers keep your eye on the endpoint of the n^{th} trailer



the midpoints of the two sets of rear wheels; it should be noted that if the trailers are hitched behind the rear axle, the equations will not simplify as shown here.

The wheels of the robot and trailers are constrained to roll without slipping; this implies that the velocity of each body in the direction perpendicular to its wheels must be zero. We model each pair of rear wheels as a single wheel at the midpoint of the axle, and state the non-slipping conditions in terms of coordinates, beginning with the n^{th} trailer:

$$0 = \dot{x}_n \sin\theta_n - \dot{y}_n \cos\theta_n. \tag{17}$$

Equation (17) models the fact that the velocity perpendicular to the wheels is zero. In the language of one forms we write this as

$$\alpha^1(x_n, y_n, \theta_n, \ldots, \theta_0) = \sin\theta_n dx_n - \cos\theta_n dy_n. \tag{18}$$

To write the other rolling constraints, we define v_i to be the magnitude of the velocity of the i^{th} trailer. The direction of motion of the $(i+1)^{st}$ trailer and consequently the direction of v_{i+1}, if its wheels are rolling without slipping, is along the direction of the hitch joining the $(i+1)^{st}$ body to the i^{th} body. Since the bodies are linked together by rigid rods, it follows that the projection of v_i onto the line of the hitch is equal to v_{i+1}. Thus, we have that

$$v_{i+1}(x) = \cos(\theta_{i+1} - \theta_i)v_i(x). \tag{19}$$

Also, we have that the velocity of the n^{th} trailer v_n is given by

$$v_n(x) = \cos\theta_n \dot{x}_n + \sin\theta_n \dot{y}_n. \tag{20}$$

In the sequel we will need to use v_n as a one form (*i.e.* we will need to use $v_n dt$) and we denote this by abuse of notation as:

$$v_n(x) = \cos\theta_n dx_n + \sin\theta_n dy_n. \tag{21}$$

We may now recursively write down the rolling without slipping constraints for all the trailers. The velocity of each trailer has a component due to the velocity v_{i+1} of the previous trailer and a component $L_{i+1}\dot{\theta}_{i+1}$ due to the rotation of the hitch. The relative geometry of this situation is illustrated in Figure 3. The component of v_{i+1} in the direction perpendicular to the wheel base is $v_{i+1}\sin(\theta_i - \theta_{i+1})$ and the component of $L_{i+1}\dot{\theta}_{i+1}$ in this direction is $L_{i+1}\dot{\theta}_{i+1}\cos(\theta_i - \theta_{i+1})$. If the i^{th} trailer rolls without slipping then we must have

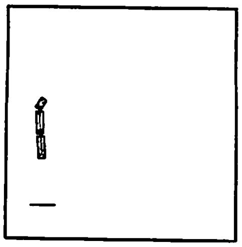
$$0 = L_{i+1}\dot{\theta}_{i+1}\cos(\theta_{i+1} - \theta_i) - v_{i+1}\sin(\theta_{i+1} - \theta_i). \tag{22}$$

Dividing through the equation (22) by $\cos(\theta_{i+1} - \theta_i)$ yields the form constraint for $n - 1 \geq i \geq 0$:

$$\alpha^{n+1-i}(x) = L_{i+1} d\theta_{i+1} - \tan(\theta_{i+1} - \theta_i)v_{i+1} = 0. \tag{23}$$

Note that we have used the one form version of v_{i+1} in equation (23).

The forms $\alpha^1(x), \alpha^2(x), \ldots, \alpha^{n+1}(x)$ represent the constraints that the wheels of the n^{th}, $(n-1)^{st}$, $\ldots$, 0^{th} trailer (*i.e.* the cab), respectively roll without slipping.



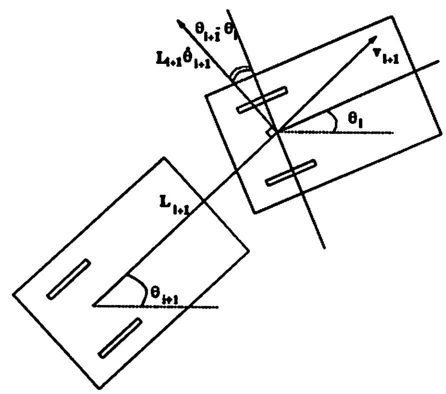


FIGURE 3. Showing the definitions of the angles and velocities of the i^{th} trailer

They are given by the formulas (23) with the recursion relations (19). Thus, the Pfaffian system for the N-trailer problem is:

$$I = \text{span}\{\alpha^1, \alpha^2, \ldots, \alpha^{n+1}\}. \tag{24}$$

The following theorem gives the derived flag associated with this Pfaffian system.

Theorem 10 (Derived Flag for the N-trailer Pfaffian system). *Consider the Pfaffian system of the N-trailer system* (24) *with the one forms α^i defined by equations* (18) *and* (23). *The one-forms α^i are adapted to the derived flag in the following sense:*

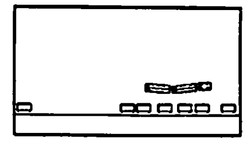
$$\begin{array}{rcl} I^{(0)} & = & \text{span}\{\alpha^1, \alpha^2, \ldots, \alpha^n, \alpha^{n+1}\} \\ I^{(1)} & = & \text{span}\{\alpha^1, \alpha^2, \ldots, \alpha^n\} \\ & \vdots & \\ I^{(n)} & = & \text{span}\{\alpha^1\} \\ I^{(n+1)} & = & \{0\}. \end{array} \tag{25}$$

Proof. The proof is by recursion starting from the bottom of the flag of (25). Indeed for the first step, we compute $d\alpha^1$ to be $d(\sin\theta_n dx_n - \cos\theta_n dy_n)$, namely:

$$\begin{aligned} d\alpha^1 &= \cos\theta_n d\theta_n \wedge dx_n + \sin\theta_n d\theta_n \wedge dy_n \\ &= (-\cos\theta_n dx_n - \sin\theta_n dy_n) \wedge d\theta_n \\ &= -v_n \wedge d\theta_n. \end{aligned}$$

From Equation (21) it follows that $d\alpha^1 \neq 0 \bmod \alpha^1$. This establishes the last two steps of the derived flag above. For the preceding step, we note that the form α^2 is given by

$$\alpha^2 = L_n d\theta_n - \tan(\theta_n - \theta_{n-1}) v_n.$$



This yields that $d\theta_n$ is proportional to v_n mod α^2. Consequently, we have that $d\alpha^1 = -v_n \wedge d\theta_n$ is equal to 0 mod α_2. This establishes that

$$\begin{aligned} I^{(n-1)} &= \operatorname{span}\{\alpha^1, \alpha^2\} \\ I^{(n)} &= \operatorname{span}\{\alpha^1\} \\ I^{(n+1)} &= \{0\}. \end{aligned} \tag{26}$$

For the i^{th} step of the recursion proof, we assume that we have shown that

$$\begin{aligned} I^{(n-i+2)} &= \operatorname{span}\{\alpha^1, \alpha^2, \ldots, \alpha^{i-1}\} \\ &\vdots \\ I^{(n)} &= \operatorname{span}\{\alpha^1\} \\ I^{(n+1)} &= \{0\}. \end{aligned} \tag{27}$$

We need to show that $d\alpha^i = 0$ mod $\alpha^1, \ldots, \alpha^{i-1}, \alpha^i$. To verify this it is useful to have the following preliminary lemma:

Lemma 11. *For the one forms v_i we have that*

$$dv_{n-i} \equiv 0 \mod \alpha^1, \alpha^2, \ldots, \alpha^{i+2}. \tag{28}$$

Proof. Start first with $dv_n = d(\cos\theta_n dx_n + \sin\theta_n dy_n)$:

$$\begin{aligned} dv_n = & \; -\sin\theta_n d\theta_n \wedge dx_n + \cos\theta_n d\theta_n \wedge dy_n \\ = & \; (\sin\theta_n dx_n - \cos\theta_n dy_n) \wedge d\theta_n \\ \equiv & \; 0 \mod \alpha^1. \end{aligned}$$

Thus $dv_n \equiv 0$ mod α^1.

From $v_{n-1} = v_n \sec(\theta_n - \theta_{n-1})$ it follows that

$$dv_{n-1} = dv_n \sec(\theta_n - \theta_{n-1}) + \sec(\theta_n - \theta_{n-1}) \tan(\theta_n - \theta_{n-1}) v_n \wedge (d\theta_n - d\theta_{n-1}).$$

The first term is zero mod α^1 since $dv_n \equiv 0$ mod α^1. The second term is zero mod α^2 since v_n is proportional to $d\theta_n$ mod α^2 and the third term is zero mod α^3 since v_n is proportional to θ_{n-1} mod α^3. Thus, we have that

$$dv_{n-1} \equiv 0 \mod \alpha^1, \alpha^2, \alpha^3.$$

Proceeding recursively, we have that

$$dv_{n-i} \equiv 0 \mod \alpha^1, \alpha^2, \ldots, \alpha^{i+2},$$

which completes the proof of the lemma. □

We will also need to make use of the relations:

$$\begin{aligned} d\theta_n &\equiv v_n && \mod \alpha^2 \\ d\theta_i &\equiv v_n && \mod \alpha^{n-i+2} \\ d\theta_{n-i+2} &\equiv v_n && \mod \alpha^i. \end{aligned} \tag{29}$$

These follow directly from the definition of the α^i in Equation (23) and the linear dependence of the one-forms v_i, given in Equation (19).

Continuing with the proof of the theorem, we now begin the calculation for

$$\begin{aligned} d\alpha^i &= d(L_{n-i+2}d\theta_{n-i+2} - \tan(\theta_{n-i+2} - \theta_{n-i+1})v_{n-i+2}) \\ &= -\sec^2(\theta_{n-i+2} - \theta_{n-i+1})(d\theta_{n-i+2} - d\theta_{n-i+1}) \wedge v_{n-i+2} \\ &\qquad - \tan(\theta_{n-i+2} - \theta_{n-i+1})dv_{n-i+2}. \end{aligned}$$

This expression has three terms. By equation (28), we have that $dv_{n-i+2} \equiv 0 \mod \alpha^1, \dots, \alpha^i$. Also by the proportionality of the $d\theta_i$ to v_n (29) and the linear dependence of the v_i's (19), we have that $d\theta_{n-i+2} \wedge v_{n-i+2} \equiv 0 \mod \alpha^i$ and $d\theta_{n-i+1} \wedge v_{n-i+2} \equiv 0 \mod \alpha^{i-1}$. Thus, we have that $d\alpha^i \equiv 0 \mod \alpha^1, \alpha^2, \dots, \alpha^i$ which implies that the derived flag has the form

$$I^{(n-i+1)} = \{\alpha^1, \dots, \alpha^i\},$$

as stated. □

We note that the $I^{(n+1)} = \{0\}$ implies that the N-trailer system is completely controllable (by Chow's theorem).

3.2. Conversion to Goursat Normal Form. In the preceding subsection, we have shown that the ideal generated by $\alpha^1, \dots, \alpha^{n+1}$ defined in equations (18) and (23) is adapted to its derived flag in the sense of (25). It remains to check whether the α^i satisfy the Goursat congruences and if they do, to find a transformation that puts them into the Goursat canonical form.

Theorem 12 (Goursat Congruences for the N-trailer system). *Consider the Pfaffian system associated with the N-trailer system* (24) *with the one-forms α^i defined by equations* (18) *and* (23), *repeated below:*

$$\begin{aligned} \alpha^1(x) &= \sin\theta_n dx_n - \cos\theta_n dy_n \\ \alpha^i(x) &= L_{n-i+2}d\theta_{n-i+2} - \tan(\theta_{n-i+2} - \theta_{n-i+1})v_{n-i+2} \\ &\qquad i = 2, \dots, n+1. \end{aligned}$$

There exists a change of basis of the one forms α^i to $\bar\alpha^i$ which preserves the adapted structure, and a one-form π which satisfies the Goursat congruences for this new basis:

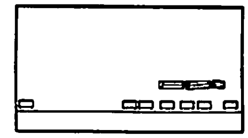
$$\begin{aligned} d\bar\alpha^i &\equiv -\bar\alpha^{i+1} \wedge \pi \mod \bar\alpha^1, \dots, \bar\alpha^i \qquad i = 1, \dots, n \\ d\bar\alpha^{n+1} &\not\equiv 0 \mod I. \end{aligned}$$

The one-form which satisfies these congruences is given by

$$\pi = \cos\theta_n dx_n + \sin\theta_n dy_n,$$

and is equivalent to v_n, the velocity form of the n^{th} trailer.

Proof. The outline for the proof is first to determine a suitable one-form π from the first Goursat congruence, $d\alpha^1 \equiv -\alpha^2 \wedge \pi$. Then we construct the new basis elements $\bar\alpha^i$ one at a time such that they satisfy the rest of the congruences. For this example, we find that these new basis elements are multiples of the original



basis elements, and since the original basis is adapted to the derived flag, the new basis is also adapted.

We determine π by following the same procedure as in Example 2. We complete the basis of $\{\alpha^1, \dots, \alpha^{n+1}\}$ with

$$\begin{aligned} \alpha^{n+2} &= \cos\theta_n dx_n + \sin\theta_n dy_n = v_n \\ \alpha^{n+3} &= d\theta_0. \end{aligned}$$

Note that $\alpha^{n+2} = v_n$, the velocity form of the last trailer. We then set $\pi = \lambda_1\alpha^{n+2} + \lambda_2\alpha^{n+3}$ and solve for λ_1, λ_2 using

$$d\alpha^1 \equiv -\alpha^2 \wedge \pi \mod \alpha^1.$$

Calculating the exterior derivative of α^1,

$$d\alpha^1 = \cos\theta_n d\theta_n \wedge dx_n + \sin\theta_n d\theta_n \wedge dy_n = d\theta_n \wedge v_n, \tag{30}$$

and then examining $\alpha^2 \wedge \pi$,

$$\alpha^2 \wedge \pi = (L_n d\theta_n - \tan(\theta_n - \theta_{n-1})v_n) \wedge (\lambda_1 v_n + \lambda_2 d\theta_0),$$

we see that if we choose $\lambda_1 = 1, \lambda_2 = 0$, then

$$\alpha^2 \wedge \pi = L_n d\theta_n \wedge v_n = L_n d\alpha^1.$$

We note here that we could have chosen $\lambda_1 = -1/L_n$, but instead we will define a new basis element $\bar{\alpha}^2 = -(1/L_n)\alpha^2$. Then the one-form $\pi = v_n$ will satisfy

$$d\alpha^1 = -\bar{\alpha}^2 \wedge \pi.$$

We now continue this procedure to find the rest of the transformed basis. Taking the exterior derivative of $\bar{\alpha}^2$,

$$\begin{aligned} d\bar{\alpha}^2 &= d(-d\theta_n + \frac{1}{L_n}\tan(\theta_n - \theta_{n-1})v_n) \\ &= \frac{1}{L_n}\sec^2(\theta_n - \theta_{n-1})(d\theta_n - d\theta_{n-1}) \wedge v_n - \frac{1}{L_n}\tan(\theta_n - \theta_{n-1})dv_n, \end{aligned}$$

and noting that

$$\begin{aligned} v_n \wedge d\theta_n &\equiv 0 \mod \bar{\alpha}^2 \\ dv_n &\equiv 0 \mod \alpha^1, \end{aligned}$$

it can be seen that

$$d\bar{\alpha}^2 \equiv -\frac{1}{L_n}\sec^2(\theta_n - \theta_{n-1})d\theta_{n-1} \wedge v_n \mod \alpha^1, \bar{\alpha}^2.$$

Also, since

$$\alpha^3 \wedge \pi = L_{n-1} d\theta_{n-1} \wedge v_n,$$

a choice of

$$\bar{\alpha}^3 = \frac{1}{L_n L_{n-1}}\sec^2(\theta_n - \theta_{n-1})\alpha^3$$

will result in the congruence:

$$d\bar{\alpha}^2 \equiv -\bar{\alpha}^3 \wedge \pi \quad \text{mod } \alpha^1, \bar{\alpha}^2.$$

Since the new basis we are defining is merely a scaled version of the original basis, mod-ing out by α^i or $\bar{\alpha}^i$ is equivalent.

Continuing, we find

$$d\bar{\alpha}^3 = d\Big(\frac{1}{L_n}\sec^2(\theta_n - \theta_{n-1})d\theta_{n-1} - \frac{1}{L_n L_{n-1}}\sec^2(\theta_n - \theta_{n-1})\tan(\theta_{n-1} - \theta_{n-2})v_{n-1}\Big). \tag{31}$$

Referring to Lemma 11 and Equation (29), we see that the following congruences hold,

$$\begin{aligned} d\theta_n \wedge v_n &\equiv 0 \ \text{mod } \alpha^2 \\ d\theta_{n-1} \wedge v_n &\equiv 0 \ \text{mod } \alpha^3 \\ d\theta_n \wedge d\theta_{n-1} &\equiv 0 \ \text{mod } \alpha^2, \alpha^3 \\ dv_{n-1} &\equiv 0 \ \text{mod } \alpha^1, \alpha^2, \alpha^3, \end{aligned}$$

and that using these, Equation (31) can be reduced to

$$\begin{aligned} d\bar{\alpha}^3 &\equiv \frac{1}{L_n L_{n-1}}\sec^2(\theta_n - \theta_{n-1})\sec^2(\theta_{n-1} - \theta_{n-2})d\theta_{n-2} \wedge v_{n-1} \ \text{ mod } \alpha^1, \bar{\alpha}^2, \bar{\alpha}^3 \\ &\equiv \frac{1}{L_n L_{n-1}}\sec^3(\theta_n - \theta_{n-1})\sec^2(\theta_{n-1} - \theta_{n-2})d\theta_{n-2} \wedge v_n \ \text{ mod } \alpha^1, \bar{\alpha}^2, \bar{\alpha}^3. \end{aligned}$$

In the second expression we have written v_{n-1} in terms of v_n. Also, $\alpha^4 \wedge \pi = L_{n-2}d\theta_{n-2} \wedge v_n$, so if we define

$$\bar{\alpha}^4 = \frac{-1}{L_n L_{n-1} L_{n-2}}\sec^3(\theta_n - \theta_{n-1})\sec^2(\theta_{n-1} - \theta_{n-2})\alpha^4,$$

then the congruence

$$d\bar{\alpha}^3 \equiv -\bar{\alpha}^4 \wedge \pi \quad \text{mod } \alpha^1, \bar{\alpha}^2, \bar{\alpha}^3$$

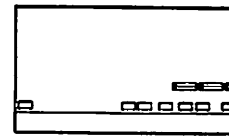
results.

In general, we assume that $\bar{\alpha}^i$ has been defined as

$$\bar{\alpha}^i = \frac{(-1)^{i-1}}{L_n \cdots L_{n-i+2}}\sec^{i-1}(\theta_{n-1} - \theta_n)\sec^{i-2}(\theta_{n-2} - \theta_{n-1})\cdots\sec^2(\theta_{n-i+3} - \theta_{n-i+2})\alpha^i.$$

Using the congruences

$$\begin{aligned} d\theta_{n-i} \wedge d\theta_{n-i+1} &\equiv 0 \ \text{mod } \alpha^{i+2}, \alpha^{i+3} \\ d\theta_{n-i} \wedge v_n &\equiv 0 \ \text{mod } \alpha^{i+2} \\ dv_{n-i} &\equiv 0 \ \text{mod } \alpha^1, \ldots, \alpha^{i+2}, \end{aligned}$$



we can show that

$$\begin{aligned} d\bar{\alpha}^i \equiv & \frac{(-1)^{i-1}}{L_n \cdots L_{n-i+2}} \sec^{i-1}(\theta_{n-1} - \theta_n) \sec^{i-2}(\theta_{n-2} - \theta_{n-1}) \cdots \\ & \sec^2(\theta_{n-i+3} - \theta_{n-i+2}) \sec^2(\theta_{n-i+2} - \theta_{n-i+1}) d\theta_{n-i+1} \wedge v_{n-i+2} \\ & \bmod \alpha^1, \bar{\alpha}^2, \ldots, \bar{\alpha}^i \\ \equiv & \frac{(-1)^{i-1}}{L_n \cdots L_{n-i+2}} \sec^{i}(\theta_{n-1} - \theta_n) \sec^{i-1}(\theta_{n-2} - \theta_{n-1}) \cdots \\ & \sec^3(\theta_{n-i+3} - \theta_{n-i+2}) \sec^2(\theta_{n-i+2} - \theta_{n-i+1}) d\theta_{n-i+1} \wedge v_n \\ & \bmod \alpha^1, \bar{\alpha}^2, \ldots, \bar{\alpha}^i \\ \equiv & - \bar{\alpha}^{i+1} \wedge v_n \mod \alpha^1, \bar{\alpha}^2, \ldots, \bar{\alpha}^i. \end{aligned}$$

All that remains now is to demonstrate that

$$d\bar{\alpha}^{n+1} \neq 0 \mod I.$$

From the above analysis, we know

$$\begin{aligned} d\bar{\alpha}^{n+1} \equiv & \frac{(-1)^n}{L_n \cdots L_1} \sec^{n+1}(\theta_{n-1} - \theta_n) \cdots \sec^3(\theta_2 - \theta_1) \sec^2(\theta_1 - \theta_0) d\theta_0 \wedge v_n \\ & \bmod \alpha^1, \ldots, \bar{\alpha}^{n+1} \end{aligned}$$

which is nonzero. □

Now that we have shown that the one-forms α^i do satisfy the Goursat congruences, we can follow the steps of the algorithm of Section 2 to find the coordinate transformation that will result in Goursat normal form. Following Algorithm 1, in step 2 we look for *possibly non-unique* functions f_1, f_2 which satisfy (10), namely

$$\begin{aligned} d\alpha^1 \wedge \alpha^1 \wedge df_1 &= 0 \\ \alpha^1 \wedge df_1 \wedge df_2 &= 0 \end{aligned} \quad \text{and} \quad \begin{aligned} \alpha^1 \wedge df_1 &\neq 0 \\ df_1 \wedge df_2 &\neq 0. \end{aligned} \tag{10}$$

Since $\alpha^1 = \sin\theta_n dx_n - \cos\theta_n dy_n$ and $d\alpha^1 = -\cos\theta_n dx_n \wedge d\theta_n - \sin\theta_n \wedge d\theta_n$, it follows that $d\alpha^1 \wedge \alpha^1 = dx_n \wedge dy_n \wedge d\theta_n$. Thus f_1 may be chosen to be *any* function of x_n, y_n, θ_n *exclusively*. We now proceed to explain two different solutions of the equations (10):

Transformation 1: Coordinates of the N^{th} trailer. In a choice motivated by Sørdalen [32] we choose $f_1 = x_n$. Then, the second equation of (10) becomes

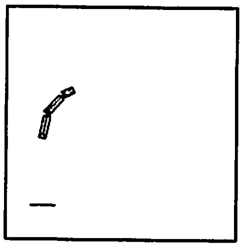
$$\sin\theta_n dx_n \wedge dy_n \wedge df_2 = 0$$

with the proviso that $df_1 \wedge df_2 \neq 0$. A *non-unique* choice of f_2 is

$$f_2 = y_n.$$

For the change of coordinates, we choose

$$\begin{aligned} z_1 = & \; f_1(x) &= x_n \\ z_{n+3} = & \; f_2(x) &= y_n. \end{aligned}$$



The one form $\alpha^1 = 0$ may be written by dividing through by $\sin\theta_n$ as

$$\begin{aligned}\alpha^1 = &\ dy_n + \tan\theta_n dx_n \\ = &\ dz_{n+3} - z_{n+2}dz_1,\end{aligned}$$

so that

$$z_{n+2} = -\tan\theta_n.$$

By the proof of Engel's theorem, we now need to find $a(x), b(x)$ so that:

$$\begin{aligned}\alpha^2 \equiv &\ a(x)dz_{n+2} + b(x)dz_1 && \mod \alpha^1 \\ = &\ -a(x)\sec^2(\theta_n)d\theta_n + b(x)dx_n && \mod \alpha^1.\end{aligned}$$

But $\alpha^2 = L_n d\theta_n - \tan(\theta_n - \theta_{n-1})v_n$. Hence, we have that

$$a(x) = \frac{-L_n}{\sec^2\theta_n} \qquad b(x) = \frac{-\tan(\theta_n - \theta_{n-1})}{\cos\theta_n},$$

and we may write

$$\alpha^2 = dz_{n+2} - \frac{-b(x)}{a(x)}dz_1.$$

Now, we define

$$z_{n+1} := -\frac{b(x)}{a(x)} = \frac{\tan(\theta_n - \theta_{n-1})\cos\theta_n}{L_n}.$$

The remaining coordinates are found by solving the equations

$$\alpha^i = dz_{n-i+4} - z_{n-i+3}dz_1 \quad \mod \alpha^1, \dots, \alpha^{i-1}$$

for $i \geq 2$. The details are not particularly insightful and are omitted here.

Transformation 2: Coordinates of the origin seen from the last trailer. Yet another choice for f_1 corresponds to writing the coordinates of the origin as seen from the last trailer. This is reminiscent of a transformation used by Samson [31] in a different context, and is given by

$$z_1 := f_1(x) = x_n \cos\theta_n + y_n \sin\theta_n.$$

This has the physical interpretation of being the origin of the reference frame when viewed from a coordinate frame attached to the n^{th} trailer. It satisfies the first of the equations of (10) simply by virtue of the fact that it is a function of x_n, y_n, θ_n. It may be verified that a choice of f_2 (non-unique—we got it by guess work!) given by

$$z_{n+3} := f_2 = x_n \sin\theta_n - y_n \cos\theta_n - \theta_n z_1$$

satisfies

$$\alpha^1 \wedge df_1 \wedge df_2 = 0.$$

The remaining coordinates $z_2, \dots, z_{n+2}$ corresponding to this transformation may be obtained from the same procedure as in the previous solution. The details are tedious.[3] In the next subsection, we discuss yet another technique for obtaining the coordinates for the Goursat normal form.

3.3. Conversion to Chained Form. In the previous section, we described a method for converting the N-trailer exterior differential system into Goursat normal form. Recalling from Section 2 that the dual of Goursat normal form is chained form, we now show how a similar procedure can be used to transform the the nonholonomic control system corresponding to the N-trailer system into chained canonical form.

We note that an exterior differential system on $\mathbb{R}^n$ of codimension 2, given by

$$I = \{\alpha^1(x), \dots, \alpha^{n-2}(x)\},$$

is dual to a two-input nonholonomic control system:

$$\Sigma: \qquad \dot{x} = g_1(x)u_1 + g_2(x)u_2, \tag{32}$$

where the vector fields $g_j(x)$ span a 2-dimensional distribution Δ which is annihilated by the one-forms α^i:

$$\alpha^i(x) \cdot g_j(x) = 0.$$

When we transform an exterior differential system into Goursat normal form, we only perform a coordinate transformation $z = f(x)$. There is no input *per se* to a formal exterior differential system, although we can speak of the two degrees of freedom of the system, given by the distribution $\Delta = I^{\perp}$.

The procedure for transforming a nonholonomic control system such as (32) into chained form requires both a coordinate transformation and state feedback. Although for the most general case, a state feedback is given by

$$\bar{u} = a(x) + b(x)u,$$

for drift-free nonholonomic systems it is easily seen that $a(x) = 0$. (If this were not the case, the state feedback would add a drift term to a drift-free system and could not result in a chained form.) The purpose of the state feedback $\bar{u} = b(x)u$ is therefore to transform the basis of the distribution Δ into chained form in the new coordinate system:

$$\begin{aligned} \bar{g}_1(z) &= \frac{\partial}{\partial z_1} + z_2\frac{\partial}{\partial z_3} + \cdots + z_{n-1}\frac{\partial}{\partial z_n} \\ \bar{g}_2(z) &= \frac{\partial}{\partial z_2}. \end{aligned} \tag{33}$$

In this section, we follow through the calculations for transforming the nonholonomic control system of the N-trailer problem into chained form. Although some of the details are very similar to what has already been presented in Section 3.2, we want to highlight the distinctions between the exterior differential and the vector field formulations of the system, and we feel that an involved discussion is merited.

[3]Readers interested in the details of the transformation may obtain it from the first author by email or regular mail.

First, we will find a basis for $\Delta = I^{\perp}$ and show that the two vector fields which we choose as the basis have a physical meaning in the multi-trailer robotic system. We then consider the problem of putting the N-trailer control system into chained form: given the two vector fields g_1, g_2 and inputs u_1, u_2, with the system defined as

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2,$$

find a coordinate transform $z = f(x)$ and an input transformation $\bar{u} = b(x)u$ such that the system

$$\dot{z} = \bar{g}_1(z)\bar{u}_1 + \bar{g}_2(z)\bar{u}_2$$

is in chained form, *i.e.* $\bar{g}_1(z), \bar{g}_2(z)$ are of the form (33). We present the two coordinate transformations which were defined in the previous section along with the required input transformations, showing that as expected they result in a chained form system, and we also demonstrate that these coordinate transformations are local diffeomorphisms.

Proposition 13. *Consider an N-trailer system with* $n+1$ *rolling constraints* $\alpha^i = 0$,

$$\begin{aligned} \alpha^1(x) &= \sin\theta_n dx_n - \cos\theta_n dy_n = 0 \\ \alpha^{n+1-i}(x) &= L_{i+1} d\theta_{i+1} - \tan(\theta_{i+1} - \theta_i) v_{i+1} = 0 \qquad i = 0, \ldots, n-1, \end{aligned}$$

where the v_i *are as specified in* (19). *A basis for the distribution* Δ *which is annihilated by these one-forms* $\{\alpha^1, \ldots, \alpha^{n+1}\}$ *is given by*

$$g_1 = \begin{bmatrix} \cos\theta_n \\ \sin\theta_n \\ \frac{1}{L_n}\tan(\theta_{n-1} - \theta_n) \\ \vdots \\ \frac{1}{L_1}\prod_{i=2}^{n}\sec(\theta_{i-1} - \theta_i)\tan(\theta_0 - \theta_1) \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Proof. For the proof, we will derive the constraints α^i in a different way than was done in Section 3.1. The set of constraints that we use here comes from the condition that the pair of wheels on the i^{th} trailer cannot slide sideways; see Figure 4 for the definition of the variables associated with the i^{th} trailer. The linear velocity of the i^{th} body is $v_i = \dot{x}_i \cos\theta_i + \dot{y}_i \sin\theta_i$ and the velocity of the trailer in the direction perpendicular to the wheels is $v_i^{\perp} = \dot{x}_i \sin\theta_i - \dot{y}_i \cos\theta_i$. The non-slipping constraint requires that $v_i^{\perp} = 0$.

As stated in the beginning of Section 3.1, (x_i, y_i) can be expressed in terms of $(x_n, y_n, \theta_n, \ldots, \theta_i)$:

$$\begin{aligned} x_i &= x_{i+1} + L_{i+1}\cos\theta_{i+1} \\ y_i &= y_{i+1} + L_{i+1}\sin\theta_{i+1} \end{aligned}$$

to determine that

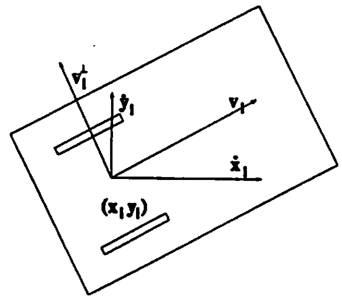


FIGURE 4. The i^{th} body, showing the velocities. The velocity v_i is defined to be in the same direction the wheels are pointing, and so the velocity perpendicular to this direction, $v_i^\perp$, must be zero if the rolling constriants are to be satisfied.

$$v_i^\perp = \dot{x}_n \sin\theta_i - \dot{y}_n \sin\theta_i - \sum_{k=i+1}^{n} L_k \cos(\theta_i - \theta_k)\dot{\theta}_k.$$

Once again we will abuse notation and also use $v_i^\perp$ to refer to the one-form:

$$v_i^\perp = \sin\theta_i dx_n - \sin\theta_i dy_n - \sum_{k=i+1}^{n} L_k \cos(\theta_i - \theta_k) d\theta_k.$$

Remark 2. The one-forms defined by these velocity constraints

$$\omega^i := v_{n-i+1}^\perp = 0, \quad i = 1, \dots, n+1$$

are also adapted to the derived flag. Indeed, since the one-forms α^i as defined in Equation (23) are adapted to the derived flag, and the relations between the ω^i and the α^i are "triangular,"

$$\omega^i = \alpha^i + \sum_{k=1}^{i-1} c_k^i \alpha^k,$$

for some coefficient functions c_k^i, it follows that the ω^i are adapted to the derived flag as well.

Because $\{\omega^1, \dots, \omega^{n+1}\}$ form an $n+1$ dimensional co-distribution Ω on T^*M, there exists a 2-dimensional distribution Δ on M which is annihilated by Ω. A basis for this distribution is given by two linearly independent vector fields g_1, g_2 which satisfy:

$$\omega^i(x) \cdot g_j(x) = 0 \qquad \forall i = 0, \dots, n, \;\; j = 1, 2.$$

Since none of the ω^i have a term $d\theta_0$, one of the vector fields in Δ can be chosen to be

$$g_2 = \frac{\partial}{\partial\theta_0}.$$

It can be verified that choosing the other vector field

$$g_1 = \cos\theta_n \frac{\partial}{\partial x_n} + \sin\theta_n \frac{\partial}{\partial y_n} + \sum_{k=1}^{n} \frac{1}{L_k} \tan(\theta_{k-1} - \theta_k) \prod_{i=k+1}^{n} \sec(\theta_{i-1} - \theta_i) \frac{\partial}{\partial \theta_k}$$

will result in $\omega^i \cdot g_1 = 0 \ \forall i$. In a more familiar notation, these two vector fields are written as

$$g_1 = \begin{bmatrix} \cos\theta_n \\ \sin\theta_n \\ \frac{1}{L_n}\tan(\theta_{n-1} - \theta_n) \\ \vdots \\ \frac{1}{L_1}\prod_{i=2}^{n}\sec(\theta_{i-1} - \theta_i)\tan(\theta_0 - \theta_1) \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

where the coordinates are written in the order $x = (x_n, y_n, \theta_n, \ldots, \theta_0)$.

Although there are many different choices of g_1, g_2 which will span Δ, the two which we have picked are natural in the sense that when the nonholonomic control system is written as:

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2$$

the input functions have the physical meanings: $u_1 = v_n$ is the linear velocity of the n^{th} trailer, and $u_2 = \omega$ is the rotational velocity of the lead car. From a practical point of view, we have control only on the velocity v_0 of the lead car given in terms of v_n by

$$v_0 = \sec(\theta_0 - \theta_1)\sec(\theta_1 - \theta_2)\cdots\sec(\theta_{n-1} - \theta_n)v_n.$$

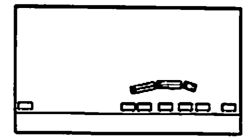
This is merely an input transformation, and will not change any of the properties of the chained form system. $\square$

We will now derive the coordinate transformations and changes of input required to put the system into chained form, as was discussed in Section 2.7. Recall that a system in chained canonical form is defined to be

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1 \\ &\vdots \\ \dot{z}_m &= z_{m-1} u_1. \end{aligned}$$

We note that the functions $z_1(t)$ and $z_m(t)$ will completely define all the state variables of a chained-form system,[4] since the other $m-2$ states and the two inputs

[4]As this paper was being finished it was pointed out to the authors that this situation is referred to by Fliess *et al.* as flat outputs [11].



can be determined from the equations:

$$\begin{aligned} u_1 &= \dot{z}_1 \\ z_i &= \dot{z}_{i+1}/u_1 \qquad i = m-1, \ldots, 2 \\ u_2 &= \dot{z}_2. \end{aligned} \tag{34}$$

Consequently, a coordinate transformation into chained form is completely defined by the first and last coordinates of the chain, z_1 and z_m, as functions of the original coordinates x along with equation (34). (The fact that such a transform exists follows from our having verified the Goursat congruences for the α^i in the previous subsection.) It does need to be checked that the transformation which results from equation (34) is a valid diffeormorphism. In general, there are many possible transformations into chained form; two are presented here. These two are exactly the same as those discussed in the previous subsection in the context of the Goursat normal form.

Transformation 1: Coordinates of the N^{th} trailer. Originally proposed by Sørdalen [32], and also used in the previous section, is as follows:

$$\begin{aligned} z_1 &= x_n \\ z_{n+3} &= y_n. \end{aligned}$$

The corresponding input transformation is:

$$\bar{u}_1 = \dot{z}_1 = \cos\theta_n v_n = \cos(\theta_0 - \theta_1)\cos(\theta_1 - \theta_2)\cdots\cos(\theta_{n-1} - \theta_n)v_0.$$

The other input $\bar{u}_2 = \dot{z}_2$ is a quite complicated function of x, v_0, ω for the general case with n trailers. However, it is easily verified that

$$\frac{\partial \bar{u}_2}{\partial \omega} \neq 0,$$

implying that the input transformation $\bar{u} = b(x)u$ is nonsingular. The remaining coordinates $z = f(x)$ are defined using equation (34); Mathematica code which generates these coordinates symbolically is given in Appendix A.

It can be checked that this coordinate transformation is valid by looking at the Jacobian,

$$[\frac{\partial z}{\partial x}] = \left[\begin{array}{cc|ccc} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & * & & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & * & & * \end{array}\right]$$

where the coordinates are written in the order: $x = (x_n, y_n, \theta_n, \theta_{n-1}, \ldots, \theta_0)$, $z = (z_1, z_{n+3}, z_{n+2}, \ldots, z_2)$ and $*$ represents any nonzero function. The ordering of the z coordinates was chosen to put the Jacobian matrix in a lower-triangular form, thereby highlighting its nonsingularity. That the Jacobian is nonsingular implies that the map $f : x \to z$ is a local diffeomorphism.

It should be noted that this coordinate transformation is only defined locally. Since its definition requires a division by u_1, if any of the factors in u_1 are zero, the transformation is undefined for that particular configuration. For example, if $\theta_n = \pi/2$, corresponding to the last trailer being at right-angles with the coordinate frame, this coordinate transformation is no longer valid. In addition, if the i^{th} trailer is *jack-knifed*, that is to say, for some $1 \leq i \leq n$, $\theta_i = \theta_{i-1} \pm \pi/2$, the coordinate transformation is also singular.

Transformation 2: Coordinates of the origin as seen from the last trailer. Another coordinate transformation which also has some singularities but will allow the trailer to be at any orientation with respect to the coordinate frame, was also detailed in the previous section in the forms version; we define it here as:

$$\begin{aligned} z_1 &= x_n \cos\theta_n + y_n \sin\theta_n \\ z_{n+3} &= x_n \sin\theta_n - y_n \cos\theta_n - \theta_n z_1. \end{aligned}$$

The input transformation and the rest of the coordinates follow from Equation (34), Mathematica code which generates both the coordinate and input transformations is given in Appendix A. Once again, it can be verified that the input transformation has the form:

$$\begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{pmatrix} = \begin{bmatrix} b_{1,1}(x) & 0 \\ b_{2,1}(x) & b_{2,2}(x) \end{bmatrix} \begin{pmatrix} v_0 \\ \omega \end{pmatrix}$$

with $b_{1,1}$ and $b_{2,2}$ nonzero functions of x. This implies that the input transformation is nonsingular.

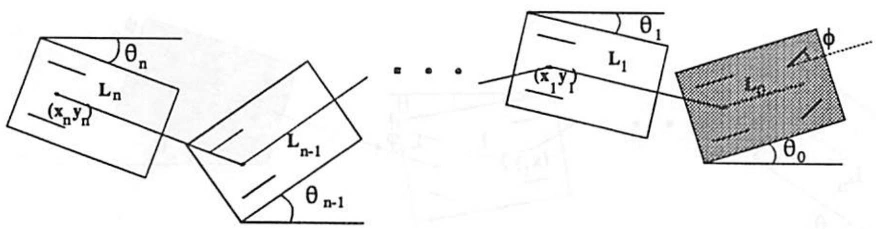
We can show that this coordinate transformation is nonsingular by looking at its Jacobian:

$$[\frac{\partial z}{\partial x}] = \left[\begin{array}{ccc|ccc} \cos\theta_n & \sin\theta_n & * & 0 & & 0 \\ \sin\theta_n & -\cos\theta_n & * & & \ddots & \\ 0 & 0 & -1 & 0 & & 0 \\ \hline 0 & 0 & * & * & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \\ 0 & 0 & * & * & & * \end{array}\right]$$

where the coordinates are written in the order: $x = (x_n, y_n, \theta_n, \theta_{n-1}, \ldots, \theta_0)$ and $z = (z_1, z_{n+3}, z_{n+2}, \ldots, z_2)$ and $*$ represents any nonzero function. Again, since the Jacobian is nonsingular, the map $f : x \to z$ is a local diffeomorphism. The singularities in this transformation also occur when division by u_1 is undefined. This happens when the expression

$$L_n + (y\cos\theta_n - x\sin\theta_n)\tan(\theta_n - \theta_{n-1}) = 0,$$

and also when any of the trailers is jack-knifed.



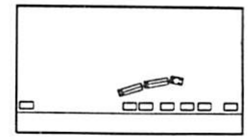


FIGURE 5. The front-wheel drive car with n trailers. This model is similar to that of Hilare 2 with an extra axle added to the front of the first body in the chain.

3.4. Generalizations. Thus far, we have concentrated our attention on the example of the Hilare mobile robot pulling a chain of trailers. In this section we demonstrate that this model is equivalent (under a coordinate transformation and state feedback) not only to the more familiar system of a front-wheel drive car pulling trailers, but also to the luggage trains commonly found in airports.

The model of the front-wheel drive car is shown in Figure 5. In comparison with the Hilare model, we have added another axle to the front body of the chain, and a variable ϕ representing the angle of the front wheels with respect to the car. The length of the wheelbase of the lead car is defined to be L_0.

The equivalence between the two models is most easily seen by looking at the form constraints. Each constraint corresponds to one axle rolling without slipping. Hilare with n trailers has $n+1$ axles; the car with n trailers has $n+2$ axles, and its Pfaffian system is therefore equivalent to that of Hilare pulling $n+1$ trailers.

Of course, the states and inputs that we define for the car system are slightly different. By convention, we define the angle of the front axle relative to the car instead of relative to the coordinate frame. This angle ϕ is merely $\theta_0 - \theta_1$ on the Hilare system. The velocity input is the same, assumed to be the linear velocity of the first body (we can define it at either the front or rear axle depending on whether our car is front-wheel drive or rear-wheel drive), but the rotational input is usually taken as $\omega' = \dot{\phi}$ the steering wheel velocity. Since in the Hilare case, we can control the velocity of the first body $\omega = \dot{\theta}_0$, state feedback can be used to reconcile these differences. As mentioned in the proof of Proposition 13, there are many choices of vector fields orthogonal to a given Pfaffian system with each choice having a different physical meaning.

The luggage carts used at most airports are also equivalent to the Hilare model. Each trailer on the luggage cart train has two sets of wheels; the front axle can spin freely about its center but the back axle is constrained to be aligned with the trailer (see Figure 6). Here again we have defined the angles of the front wheels with respect to each trailer, but looking at the form constraints it is easily seen that the cab with n luggage trailers is equivalent to a front-wheel drive car with $2n$ one-axle trailers. Again, a coordinate transformation is needed, since in the model of the luggage carts we define the angle of the front wheels of the trailers relative to the trailer instead of relative to the coordinate frame.

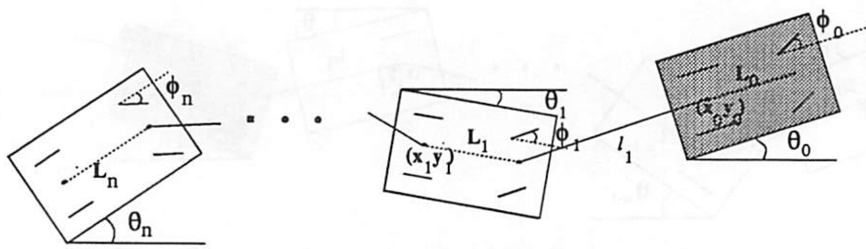


FIGURE 6. A car pulling n luggage carts. Each trailer has two axles; the front axle is free to spin about its midpoint but the rear axle is constrained to be aligned with the body of the trailer.

4. Steering Chained Form Systems

Now that we have seen how to transform an N-trailer system into chained form, we examine various methods for steering chained form systems:

$$
\begin{aligned}
\dot{z}_1 &= u_1 \\
\dot{z}_2 &= u_2 \\
\dot{z}_3 &= z_2 u_1 \\
&\vdots \\
\dot{z}_m &= z_{m-1} u_1 .
\end{aligned} \tag{35}
$$

We assume an m-state system, and note that Hilare with n trailers has $n+3$ states, a car with n trailers has $n+4$ states, and a car with n luggage trailers has $2n+4$ states.

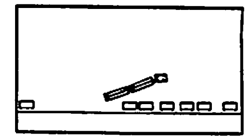
The problem that we address in this section is: Given a system in chained form with an initial state z^0 and a goal state z^f, find some control inputs $u_1(t), u_2(t)$ which will steer the system from z^0 to z^f after some time T. The application of these results to the problem of steering the mobile robot with multiple trailers is covered in the next section.

We present three methods to steer the chained form system:

(1) Sinusoidal inputs
(2) Piecewise constant inputs
(3) Polynomial inputs

4.1. Sinusoidal inputs. The first steering method that we consider uses sinusoidal inputs. Steering chained form systems with sinusoids was originally proposed by us in [29]. The method that we have developed here is different from the original algorithm in that it steers all the states in one step, instead of one state at a time.

Given an m-state chained form system, it is easily seen that the first two states, z_1 and z_2, can be steered from their initial to their final positions using constant inputs over any time period T. Of course, the states $z_3, \ldots, z_m$ will drift as a consequence of this.



By direct integration, it may be verified that a combination of out of phase sinusoids applied to the inputs,

$$u_1(t) = \alpha \sin \omega t \qquad u_2(t) = \beta \cos \omega t$$

over one period $T = 2\pi/\omega$, will cause a motion in the z_3 variable as follows:

$$\begin{aligned} z_1(T) &= z_1(0) \\ z_2(T) &= z_2(0) \\ z_3(T) &= z_3(0) + \frac{\alpha\beta}{2\omega}. \end{aligned}$$

The states $z_4, \ldots, z_m$ will drift in some fashion. Further, using inputs with u_2 having k times the frequency of u_1, namely:

$$u_1(t) = \alpha \sin \omega t \qquad u_2(t) = \beta \cos k\omega t$$

applied over one period $T = 2\pi/\omega$, will result (as may be verified directly by integration) to be

$$\begin{aligned} z_1(T) &= z_1(0) \\ &\vdots \\ z_{k+1}(T) &= z_{k+1}(0) \\ z_{k+2}(T) &= z_{k+2}(0) + \frac{\alpha^k\beta}{k!(2\omega)^k}. \end{aligned}$$

The intuition behind this steering scheme lies in the different levels of Lie brackets. If we consider the input vector fields g_1, g_2, we note that $[g_1, g_2] = [0\ 0\ 1\ 0 \cdots 0]^T$, precisely in the z_3 direction. Motion in this first level Lie bracket is generated by cycling between the two input vector fields in a continuous manner described by the out of phase sinusoids. To get motion in the second level Lie bracket, $[g_1, [g_1, g_2]] = [0\ 0\ 0\ 1\ 0 \cdots 0]^T$ or equivalently the z_4 direction, the input u_2 completes two cycles for one cycle on u_1. More generally, motion in the $\mathrm{ad}_{g_1}^k g_2 = [0 \cdots 1 \cdots 0]^T$ or the z_{k+2} direction is achieved by using k times the frequency of u_1 on u_2.

The Murray and Sastry steering algorithm is step-by-step: It first steers z_1, z_2 to their final position using constant inputs, disregarding the other states. Then it steers z_3 to its desired final position using sinusoids, z_1, z_2 will return to their final values. Now z_4 can be steered, and similarly on down the chain, until all states are at their final positions. This is a simple algorithm that is easy to implement, but can be time-consuming when there are many states to be steered.

We propose instead an "all-at-once" sinusoids method, combining all the frequencies on u_2 together in one step,

$$\begin{aligned} u_1 &= a_0 + a_1 \sin \omega t \\ u_2 &= b_0 + b_1 \cos \omega t + b_2 \cos 2\omega t + \cdots + b_{m-2} \cos(m-2)\omega t. \end{aligned} \tag{36}$$

It is no longer as simple to choose appropriate values for the parameters $(a_0, a_1, b_0, \ldots, b_{m-2})$ because of the drift that we were able to ignore when we considered each state individually. However, it is still possible to integrate the chained form equations sequentially, finding $z_1(t), z_2(t), z_3(t), \ldots, z_m(t)$ which result from the inputs (36) above. The state $z(t)$ is a function of the initial condition z^0 as well as the input parameters $a_0, a_1, b_0, \ldots, b_{m-2}$. If we evaluate $z(T)$, with $T = 2\pi/\omega$, all the sinusoidal functions will evaluate to either 0 or 1. By setting $z(T) = z^f$ we get a set of m polynomial equations in the $(m+1)$ input parameters $(a_0, a_1, b_0, \ldots, b_{m-2})$. The following proposition guarantees the existence of solutions to these equations at least locally around z^0.

Proposition 14. *Consider an m-state chained form system with initial and final states z^0, z^f. If $|z^0 - z^f| < \delta$ small, then there exist input parameters $(a_0, a_1, b_0, \ldots, b_{m-2})$ such that the inputs*

$$\begin{aligned} u_1 &= a_0 + a_1 \sin \omega t \\ u_2 &= b_0 + b_1 \cos \omega t + b_2 \cos 2\omega t + \cdots + b_{m-2} \cos(m-2)\omega t \end{aligned}$$

will steer the system from z^0 to z^f in time $T = 2\pi/\omega$.

Proof. Consider the map

$$\phi_{z^0} : \mathbb{R}^m \to \mathbb{R}^m$$

which takes values in the parameter space $(a_0, b_0, \ldots, b_m)$ and maps them to values in the state space $(z_1^f, \ldots, z_m^f)$. We define $\phi_{z^0}(a_0, b_0, \ldots, b_m)$ to be the value of $z(T)$ when the chained form system (35) is integrated starting at the initial condition z^0 and applying the inputs (36) over the time period $[0, T]$. We choose $a_1 \neq 0$. We will show that ϕ_{z^0} is a local diffeomorphism by looking at its directional derivatives, and demonstrating that the Jacobian of ϕ_{z^0} is nonsingular.

Let $\{e_i\}_{i=1}^m$ be the standard basis for $\mathbb{R}^m$ and let ϵ be small. Set $a_1 \neq 0$. Now consider the input parameterized by ϵe_1,

$$u_1 = \epsilon + a_1 \sin \omega t \qquad u_2 = 0.$$

Integrating the chained form equations and evaluating it at time T will give

$$\phi_{z^0}(\epsilon e_1) = z^0 + [\epsilon T \;\; 0 \;\; o(\epsilon) \cdots o(\epsilon)]^T$$

where $o(\epsilon)$ represents terms that are of linear and higher order in ϵ.

Now consider the input parameterized by ϵe_2

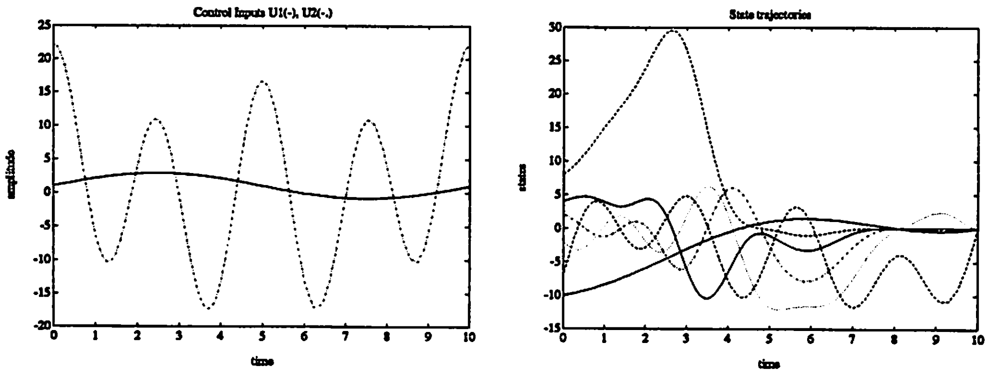
$$u_1 = a_1 \sin \omega t \qquad u_2 = \epsilon.$$

We integrate and evaluate at T as before,

$$\phi_{z^0}(\epsilon e_2) = z^0 + [0 \;\; \epsilon T \;\; o(\epsilon) \cdots o(\epsilon)]^T.$$

In this case it may be verified that $o(\epsilon)$ terms are linear in ϵ. In general, for an input parameterized by ϵe_k,

$$u_1 = a_1 \sin \omega t \qquad u_2 = \epsilon \cos(k-2)\omega t,$$



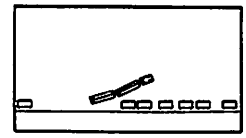


FIGURE 7. The inputs and state trajectories for a six-state, chained form system, steering from (-10,-7,-2,2,4,8) to the origin. The input u_1 is sinusoidal of one period; u_2 is a sum of sinusoids, of which the highest frequency is 4ω.

the directional derivative of ϕ in the e_k direction is given by:

$$\phi_{z^0}(\epsilon e_k) = z^0 + [0 \cdots 0 \;\; p(\epsilon) \;\; o(\epsilon) \cdots o(\epsilon)]^T,$$

where

$$p(\epsilon) = \frac{a_1^{k-2}\epsilon}{(k-2)!(2\omega)^{k-2}}.$$

These m directional derivatives are seen to be linearly independent; implying that the Jacobian of ϕ_{z^0} is nonsingular, and that ϕ_{z^0} is a local diffeomorphism. □

Remark 3. We have dealt with the overparameterization of the input ($m+1$ parameters: $a_0, a_1, b_0, \ldots, b_{m-2}$ and m states) by initially choosing a value for a_1 and then solving the m equations for the remaining m input parameters.

We note here that by choosing a fixed value for a_1, we are requiring u_1 to go through one period. Since u_1 roughly corresponds to the driving input in a mobile robot system, paths planned using the sinusoidal method generally have one back-up or speed reversal, corresponding to the zero-crossing of u_1. Parallel-parking type maneuvers seem particularly well-suited to sinusoidal trajectories.

Remark 4. Appendix A contains Mathematica code which symbolically integrates the chained form system and solves for the input parameters $a_0, b_0, \ldots, b_{m-2}$ in terms of a_1, ω, and the initial and final states z^0, z^f.

A sample of the input functions and state trajectories for a sinusoidal steering problem is shown in Figure 7. There are six states, in chained form, steering from an initial position of $(z_1, z_2, z_3, z_4, z_5, z_6) = (-10, -7, -2, 2, 4, 8)$ to the origin. The parameters were chosen to be $T = 10$ seconds and $a_1 = \frac{6\pi}{T}$.

4.2. Piecewise Constant Inputs. The second method we investigate for steering chained form systems uses piecewise constant inputs. This method was originally proposed by Monaco and Normand-Cyrot [26], and was inspired by multirate digital control. It is most easily understood in the context of nonholonomic motion planning simply as piecewise constant inputs.

Consider holding the inputs u_1 and u_2 constant over some small time period $[0, \bar{\delta})$,

$$\begin{aligned} u_1(\tau) = &\ u_{1,1} \\ u_2(\tau) = &\ u_{2,1}. \end{aligned} \qquad \tau \in [0, \bar{\delta})$$

The chained form state equations can then be integrated, and evaluated at time $\bar{\delta}$ to yield

$$\begin{aligned} z_1(\bar{\delta}) &= z_1(0) + u_{1,1}\bar{\delta} \\ z_2(\bar{\delta}) &= z_2(0) + u_{2,1}\bar{\delta} \\ z_3(\bar{\delta}) &= z_3(0) + z_2(0)u_{1,1}\bar{\delta} + u_{1,1}u_{2,1}\frac{\bar{\delta}^2}{2} \\ &\ \vdots \\ z_m(\bar{\delta}) &= z_m(0) + z_{m-1}(0)u_{1,1}\bar{\delta} + \cdots + u_{2,1}u_{1,1}^{m-2}\frac{\bar{\delta}^{m-1}}{(m-1)!}. \end{aligned} \tag{37}$$

We can now consider another pair of constant inputs on the time interval $[\bar{\delta}, 2\bar{\delta})$,

$$\begin{aligned} u_1(\tau) = &\ u_{1,2} \\ u_2(\tau) = &\ u_{2,2}. \end{aligned} \qquad \tau \in [\bar{\delta}, 2\bar{\delta})$$

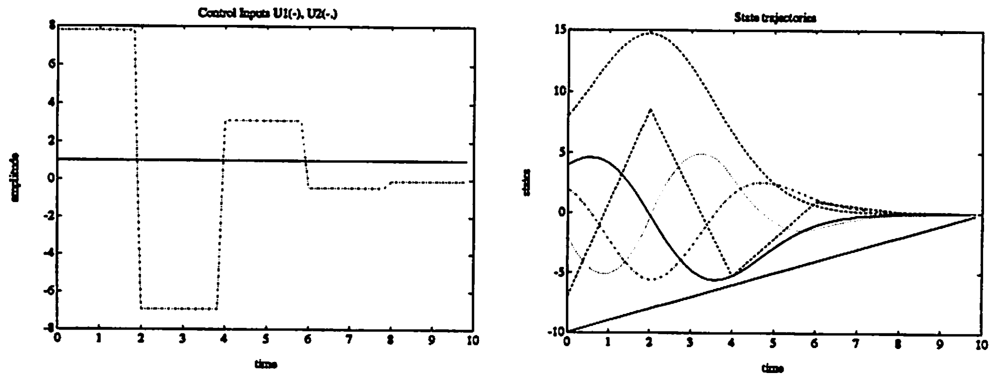
Integration of the state equations gives us $z(2\bar{\delta})$ as a function of $z(\bar{\delta})$, $u_{1,2}$, $u_{2,2}$. Using $z(\bar{\delta})$ from equation (37), we get an expression for $z(2\bar{\delta})$ in terms of $z(0)$, $u_{1,1}$, $u_{1,2}$, $u_{2,1}$, $u_{2,2}$. This procedure of piecewise integration and substitution can be repeated as many times as necessary.

For path planning, we choose to keep u_1 at a constant value over the entire trajectory. We therefore iterate the equations (37) $m-1$ times so as to have exactly m parameters for which to solve: $u_1, u_{2,1}, \ldots, u_{2,m-1}$. The total time needed for steering is $\delta = (m-1)\bar{\delta}$. Although δ can be chosen arbitrarily, a smaller time δ will result in larger inputs u to achieve the same path.

The m equations which result from setting $z(0) = z^0$ and $z(\delta) = z^f$ are polynomial (of order $m-2$) in u_1 but are linear in $u_{2,1}, \ldots, u_{2,m-1}$. Since u_1 is easily determined from

$$u_1 = \frac{z_1^f - z_1^0}{\delta}$$

the remaining $m-1$ linear equations can be solved for u_2 quite easily. This is one of the reasons that we propose keeping u_1 constant over the entire trajectory; if u_1 varied, we would need to solve high-order polynomial equations in the $u_{1,k}$ parameters.



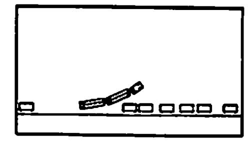


FIGURE 8. Sample inputs and state trajectories for steering a six-state chained form system with Piecewise Constant inputs. The initial position is (-10,-7,-2,2,4,8) and the goal point is the origin.

Remark 5. Appendix A contains Mathematica code which will symbolically integrate the chained form equations, and solve for the input values $u_1, u_{2,k}$ in terms of the initial and final states z^0, z^f. The code is written for a six-state system, but the generalization to any number of states is straightforward.

It should be noted that if $z_1^f = z_1^0$, or the initial and final states agree in the first coordinate, this method as stated so far will fail to yield a solution. From looking at the chained form equations, it is obvious that if $u_1 = 0$, only the second state z_2 can move; all other states must remain stationary. In practice, this case is dealt with by planning two paths, the first of which takes the initial condition to an intermediate state, the second of which joins the intermediate state with the goal position. The concatenation of these two paths is a valid trajectory between the start and goal. Our algorithm chooses the intermediate point z^m halfway between the initial and final points in all coordinates except the first, which we choose to be offset from the starting position by a constant amount,

$$\begin{aligned} z_k^m &= (z^f - z_k^0)/2, \qquad k = 2, \ldots, m \\ z_1^m &= z_1^0 + \text{const.} \end{aligned}$$

The constant offset can be adjusted to fit the situation.

The procedure detailed in the previous paragraph is used when a parallel-parking trajectory is desired for the mobile robot with trailers, since the z^1 direction in chained form corresponds to "sideways" in the original coordinates. We have found it practical to choose the constant offset at approximately twice the length of the entire robot and trailer system. A smaller offset will result in tighter turns and more lateral motion. If there are obstacles in the field, this constant offset gives a parameter that can be adjusted in an effort to avoid collisions.

Another reason for choosing u_1 to be constant over the entire trajectory is that in the mobile robot and trailer system, this input is roughly equivalent to the driving velocity. Because of the coordinate transformation that maps u_1 to the actual velocity v_0, the actual velocity of the robot will not be constant, but in most cases it will not cross zero and change sign. This means that the robot will not have to execute backing-up maneuvers to achieve its final goal position.

The main drawback of the piecewise constant inputs is the discontinuity of u_2. The models used in this paper are purely kinematic using as inputs the driving and steering velocities. In a real robot system, the inputs are not velocities but accelerations, or torques. When a path satisfying the velocity constraints is found, the input velocities need to be differentiated to find the corresponding accelerations. Of their very nature, the piecewise constant trajectories are not differentiable at the switching points.

4.3. Polynomial inputs. Yet another possibility for steering systems in chained form is to use polynomial inputs:

$$\begin{aligned} u_1 &= 1 \\ u_2 &= c_0 + c_1 t + \cdots + c_{m-2} t^{m-2}. \end{aligned}$$

This approach has the advantage of a constant input on u_1 with the added advantage of the differentiability of u_2.

The time needed to steer the system from z^0 to z^f is determined by the change desired in the first coordinate,

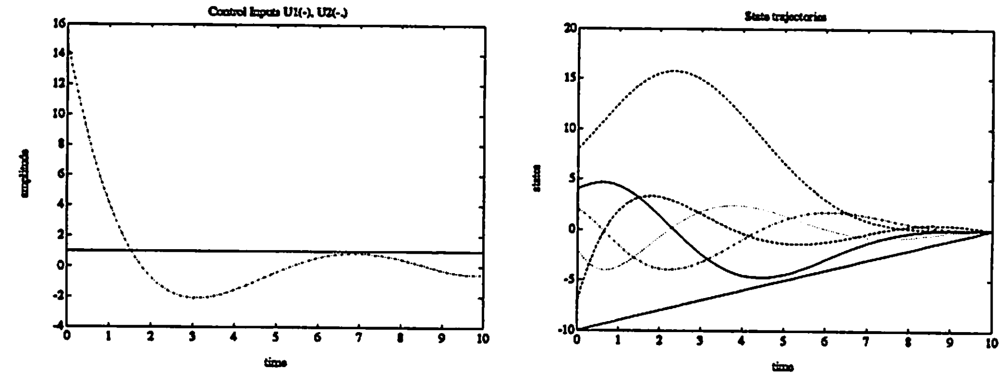
$$T = z_1^f - z_1^0.$$

Once T has been found, the state equations (35) can be integrated using the initial condition $z(0) = z^0$,

$$\begin{aligned} z_1(t) &= z_1(0) + t \\ z_2(t) &= z_2(0) + c_0 t + \frac{c_1 t^2}{2} + \cdots + \frac{c_{m-2} t^{m-1}}{m-1} \\ &\vdots \\ z_i(t) &= z_i(0) + \sum_{k=0}^{m-2} \frac{k! c_k t^{i+k-1}}{(i+k-1)!} + \sum_{k=2}^{i-1} \frac{t^{i-k}}{(i-k)!} z_k(0) \\ &\vdots \end{aligned}$$

Evaluating the foregoing at time T and setting $z(T) = z^f$ yields a total of $m-1$ equations affine in the $m-1$ variables $c_0, \ldots, c_{m-2}$,

$$M(T) \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-2} \end{bmatrix} + f(z(0), T) = \begin{bmatrix} z_2^f \\ z_3^f \\ \vdots \\ z_m^f \end{bmatrix}$$



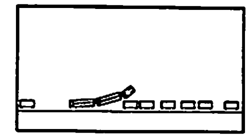


FIGURE 9. Sample trajectories and input traces for steering with Polynomial inputs. The initial position is (-10,-7,-2,2,4,8) and the goal point is the origin.

where the matrix entries $M_{i,j}(T)$ have the form:

$$M_{i,j} = \frac{(j-1)!\, T^{i+j-1}}{(i+j-1)!}.$$

It may be shown that this matrix is nonsingular for $T \neq 0$.

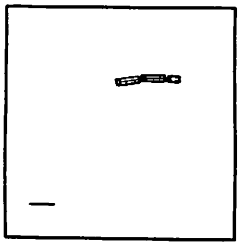
Note that if $z_1^f - z_1^0 < 0$, then we get a solution which gives a negative time period. This situation is easily remedied by choosing $u_1 = -1$ (see Appendix A for the Mathematica code which solves this problem).

As in the case of steering with piecewise constant inputs, this method will yield no solution when $z_1^f - z_1^0 = 0$. We follow the same procedure outlined in Section 4.2 to deal with this case.

4.4. Other choices. Because of the simple form of the chained form system, many different classes of input functions other than the three described above could be used to steer systems in this form. The chief requirement is that there should be at least as many parameters in the input functions as there are states. For multi-trailer systems, a desirable characteristic of the input functions is that u_1 have few or no zero-crossings since these will correspond to fewer backups. In fact, the number of backups needed to complete a manoeuvers may be taken as a measure of complexity of an input class.

5. SIMULATIONS AND OBSERVATIONS

We now have an extensive toolbox from which to choose for steering an N-trailer system. With two different coordinate transformations into chained form, and at least three different methods for steering the system once it is in chained form, we can try to pick the best combination of coordinate transformation and input type for each start and goal point. There is as yet no formal way to define when one path is "better" than another, but as we mentioned earlier, we tend to think of desirable



paths as those that have few backups and do not stray too far from the vicinity of the start and goal points.

One of the things that must be considered is coordinate singularities. Although we have shown that all three methods proposed here will find a path between any start and goal points in the chained form coordinates, there is no guarantee that this path, when transformed back into the actual coordinates, will avoid the transformation singularities. This must be checked for each desired path. If a singularity does result, another steering method might yield a valid path, or perhaps an intermediate point will need to be chosen, and the path planned in two or more steps.

In Figures 10 and 11, we show two different paths for a front-wheel drive car with two trailers. We have chosen the wheelbase of the car to be $L_1 = 0.5$ units, and each trailer to have a length of $L_2 = L_3 = 2$ units. Each path was generated using techniques described in this paper: first, transforming the start and goal points into the chained form coordinates; second, steering the chained form system using one of the methods from Section 4; and finally, transforming the trajectory back into the original coordinates.

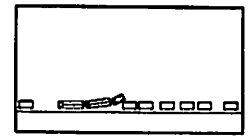
The trajectory shown in Figure 10 represents the truck backing into a loading dock. The initial condition is $(x_3, y_3, \theta_3, \theta_2, \theta_1, \theta_0) = (10,10,0,0,0,0)$ and the final position is $(0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2})$. Coordinate transformation 2 is used, since the first coordinate transformation is singular at the goal position. In the figure, we have presented the trajectory of the front of the car (x_0, y_0) instead of the back of the second trailer (x_3, y_3) to amplify the difference between the two steering methods; the trajectories of the second trailer are virtually identical.

In Figure 11 we again have chosen to present the path taken by the front car. Here we have used two different coordinate transformations with the same steering method. The trajectories in the chained form coordinates are identical; however, a difference can be seen in the physical coordinates. Once again, the trajectory traced by the rear of the second trailer is very similar in both cases. Some scenes from a movie animation of this trajectory are shown in Figure 12; in the movie we present the coordinates derived from transformation 1.

With the sinusoidal steering method, there is one parameter that can be adjusted independently of the start and goal positions; this is the magnitude of the sinusoid on the first input, or a_1 in the terminology of Section 4.1. In constructing this movie, we examined several different values of a_1; a larger value of a_1 will correspond to the car driving out farther before it starts backing into the space. We were able to choose a value for this parameter so that the car and trailer system did not hit any of the obstacles along its path.

6. Summary and Future Work

In this paper we applied the machinery of exterior differential systems to the N-trailer problem. We showed that the multi-trailer system could be put into Goursat normal form, and that this is the dual to chained form. We solved the motion planning problem for the mobile robot pulling n trailers by converting the kinematic equations into chained form, and steering the chained form system from and initial



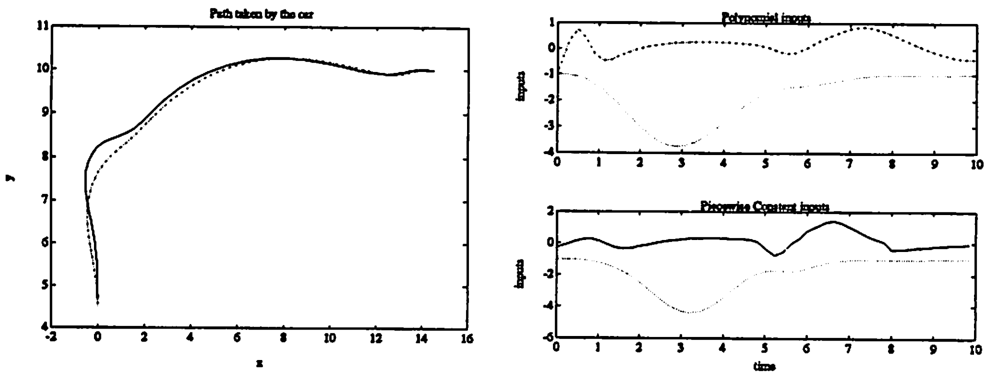


FIGURE 10. Backing a car with two trailers into a loading dock. We show here trajectories found by two different steering methods for the same initial and final conditions. The solid line corresponds to the piecewise constant inputs and the dashed line to the polynomial inputs. The x, y trace of the front of the car is shown, since the trajectory of the rear trailer is virtually identical in the two cases. Both trajectories use the second coordinate transformation. The input v_0 is the dotted line in both graphs. Clips from a movie simulation of this trajectory can be seen in Figure 13 and the movie can be viewed on the left hand pages of this paper.

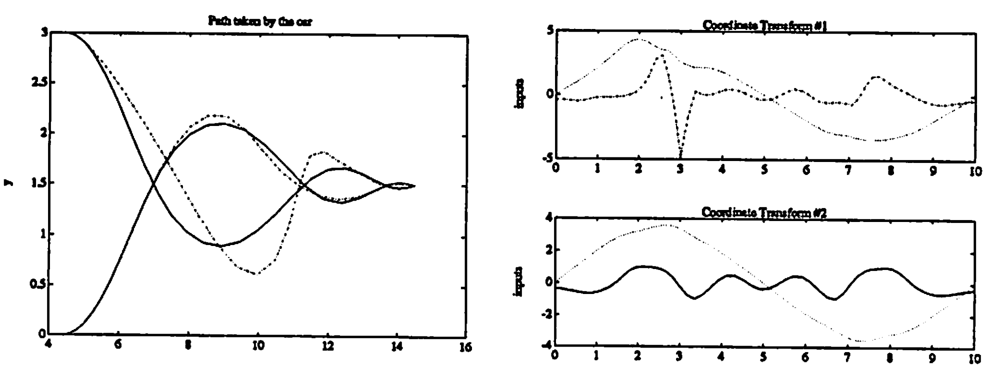
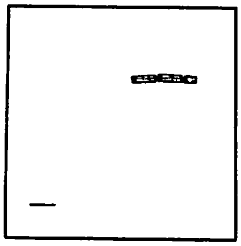


FIGURE 11. Parallel-parking a car with two trailers using sinusoids. the trace of the front car is shown for two different choices of coordinates: Transformations 1 (solid line) and 2 (dashed line). We also see how the steering input differs on with the two transformations, although for this path, the driving input v_0 (dotted line) is similar in both cases.



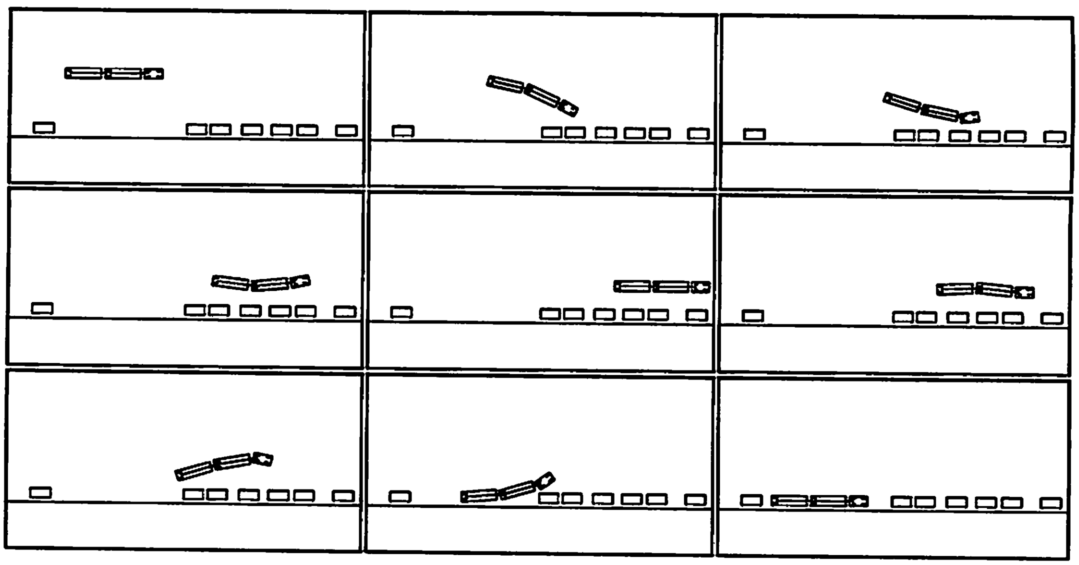
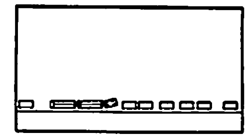


FIGURE 12. Scenes from a movie animation, showing the front-wheel drive car with two trailers (a six-state system) parallel-parking in the presence of obstacles. Sinusoidal inputs were used for steering, and the magnitude of the periodic part of the driving input (a_1 in the terminology of Section 4) was adjusted so that the obstacles were avoided. The first coordinate transformation was used. The entire movie animation can be seen on the right-hand pages of this paper.



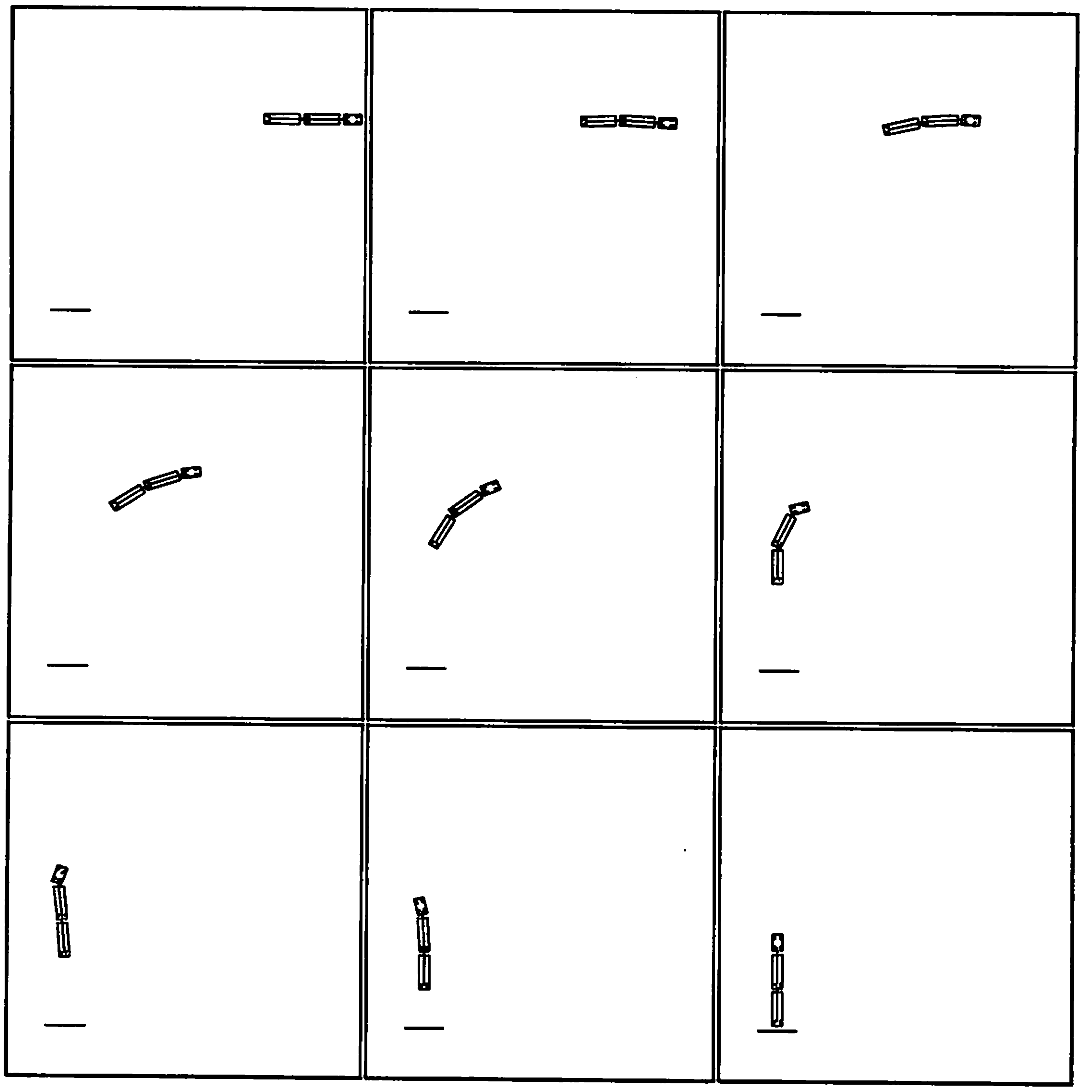
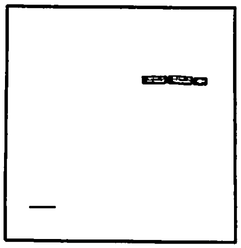


FIGURE 13. These are scenes from a movie animation, showing the front-wheel drive car with two trailers backing into a loading dock. Piecewise constant inputs were used to steer the chained form system. The entire movie simulation can be viewed on the left-hand pages of this paper.



to a final position, then converting the trajectory back into the original coordinates. Three different methods for steering chained form systems were proposed.

The work done in this paper has several natural avenues of continuation:

(1) The generation of trajectories for the N-trailer system in an environment cluttered with obstacles. This line of work has been started in [25] where the authors considered the motion of a single Hilare-like robot in a cluttered environment. Another approach for a Hilare-like robot was defined in [23]. Other methods for obstacle avoidance which use optimization based approaches may be found in [8, 9].

(2) The stabilization of open loop trajectories. The trajectories generated by our method need to be stabilized, perhaps using a technique such as that outlined in [36]. There has also been considerable interest in stabilizing nonholonomic systems not to trajectories but to points. Although from Brockett's necessary condition [3] it follows that such stabilizing control laws cannot be both C^0 and time-invariant, methods using either discontinuous or time-varying feedback have been suggested. One approach to stabilizing chained form nonholonomic systems is given in [34].

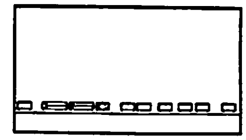
(3) Generalized Goursat type canonical forms for exterior differential systems for higher codimension systems are discussed in [12, 27] and were useful in transforming to chained form a firetruck system. The firetruck has three inputs: driving and steering in the front and another steering wheel at the tiller [5] (multi-input chained form systems are also discussed in [29]). This work is as yet far from complete since there is a very large number of different possibilities for the normal form in this instance.

(4) There are several examples of nonholonomic systems whose constraints fail to meet the conditions of the Goursat normal form, for example, the system modeling a circular finger tip rolling on a planar face [29]. The problem of steering such systems remains an open one.

ACKNOWLEDGEMENTS

We would like to thank several people who have helped us in understanding this problem over the past few years: Roger Brockett, Linda Bushnell, John Canny, Robert Gardner, Georges Giralt, Gerardo Lafferiere, Jean-Paul Laumond, Zexiang Li, Brian Mirtich, Richard Montgomery, Dorothée Normand-Cyrot, Willem Sluis, William Shadwick, Hector Sussmann and Greg Walsh. Also Sørdalen for furnishing us with his transformations of the N-trailer systems into chained form coordinates. For the margin movies we thank Anita Flynn, Joe Jones and Patrick O'Donnell of the MIT AI lab.

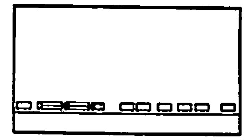
D. Tilbury and S. Sastry would like to thank Sanjoy Mitter for his hospitality at LIDS and CICS at MIT where some of this research was done in the Fall of 1992.



APPENDIX A. MATHEMATICA CODE

```
(* 3trailer.m    D. Tilbury    November 30,1992
* Defining the coordinate transformation for the system of Hilare
* with three trailers (equivalent to a car with 2 trailers).
*)

(* Definitions of the gradient, Jacobian, and the state vector *)
grad[x_,vect_] := Table[D[x,vect[[jj]]],jj,Length[vect]];
Jac[x_, vect_] := Table[D[x[[i]],vect[[j]]],i,Length[x],j,Length[vect]];
q = {x,y,th3,th2,th1,th0};

(* The input vector fields g_1 and g_2 *)
g1 ={Cos[th3], Sin[th3], Tan[th2-th3]/L3, Tan[th1-th2] Sec[th2-th3]/L2,
Tan[th0-th1] Sec[th1-th2] Sec[th2-th3]/L1 , 0 };
g2 = { 0, 0, 0, 0, 0, 1 };

(* The derivative of the state:  the two inputs here are vn, the linear
* velocity of the last body, and omega, the rotational velocity of the
* first body (Hilare)
*)
dq = g1 vn + g2 omega;

(* First define z1 and z6, the first and last chained form coords.  *)
(* Coordinate Transform 1 *)
z1 = x;
z6 = y;

(* Coordinate Transform 2
z1 = x Cos[th3] + y Sin[th3];
z6 = x Sin[th3] - y Cos[th3] - th3 z1; *)

(* u1 is then the derivative of z1, and we coax Mathematica to apply
* a useful trig identity *)
u1 = grad[z1,q].dq /.  a_ Sin[x_]^2 + a_ Cos[x_]^2 -> a;

(* Define z5, z4, z3, z2 from the derivatives of z6, z5, z4, z3 *)
z5 = Together[Expand[grad[z6,q].dq/u1]];
z4 = Factor[grad[z5,q].dq/u1];
z3 = Together[grad[z4,q].dq/u1];
z2 = Together[grad[z3,q].dq/u1];

(* then find u2 = dz2 *)
u2 = grad[z2,q].dq;
```

```
(* sinusoids.m     D. Tilbury     November 17, 1992
* Here we take a 6-state system in single-chained, 2-input form
* and steer it with sinusoids in one step (over one period).
*)

(* Set up the initial and final points *)
x0 = {x01,x02,x03,x04,x05,x06};
xf = {xf1,xf2,xf3,xf4,xf5,xf6};

(* Choose the inputs to be sums of sinusoids *)
u1[t_] := a0 + a1 Sin[w t];
u2[t_] := b0 + b1 Cos[w t] + b2 Cos[2 w t] + b3 Cos[3 w t]
+ b4 Cos[4 w t];

(* Calculate the states as functions of time *)
ete[k_] := Expand[k, Trig->True];
x1[t_] := x0[[1]] + Integrate[u1[tau],tau,0,t];
x2[t_] := x0[[2]] + Integrate[u2[tau],tau,0,t];
x3[t_] := x0[[3]] + Integrate[ete[u1[tau] x2[tau]],tau,0,t]
x4[t_] := x0[[4]] + Integrate[ete[u1[tau] x3[tau]],tau,0,t]
x5[t_] := x0[[5]] + Integrate[ete[u1[tau] x4[tau]],tau,0,t]
x6[t_] := x0[[6]] + Integrate[ete[u1[tau] x5[tau]],tau,0,t]

(* Set the time period to be T *)
T = 2 Pi/w;

(* The coefficients a0, b0, and b1 can be solved for individually.
* Due to interference, the coefficients b2, b3, and b4 must be
* found together.  All of these coefficients will be in terms of
* w and a1, both of which are free.
*)
Solve[x1[T]==xf1,a0] >>> ss_inputs.m
Solve[x2[T]==xf2,b0] >>> ss_inputs.m
Solve[x3[T]==xf3,b1] >>> ss_inputs.m
Solve[{x4[T]==xf4, x5[T]==xf5, x6[T]==xf6},{b2,b3,b4}] >>> ss_inputs.m
```



```
(* Piecewise.m      D. Tilbury     November 17, 1992
* Here we take a 6-state system in single-chained, 2-input form
* and steer it with Piecewise constant controls over 5 steps,
* d = deltabar = delta/5
*)

(* First calculate the states after one step, using constant controls
*)
x1[i_] := x1[i-1] + d u1[i];
x2[i_] := x2[i-1] + d u2[i];
x3[i_] := x3[i-1] + d x2[i-1] u1[i] + d^2 u2[i] u1[i]/2;
x4[i_] := x4[i-1] + d x3[i-1] u1[i] + d^2 x2[i-1] u1[i]^2/2 +
d^3 u1[i]^2 u2[i]/3!;
x5[i_] := x5[i-1] + d x4[i-1] u1[i] + d^2 x3[i-1] u1[i]^2/2 +
d^3 x2[i-1] u1[i]^3/3!  + d^4 u1[i]^3 u2[i]/4!;
x6[i_] := x6[i-1] + d x5[i-1] u1[i] + d^2 x4[i-1] u1[i]^2/2 +
d^3 x3[i-1] u1[i]^3/3!  + d^4 x2[i-1] u1[i]^4/4!  +
d^5 u1[i]^4 u2[i]/5!;

(* Then we iterate for 5 steps, *)
x1T = Expand[x1[5]];
x2T = Expand[x2[5]];
x3T = Expand[x3[5]];
x4T = Expand[x4[5]];
x5T = Expand[x5[5]];
x6T = Expand[x6[5]];

(* We define u1 to be constant over all 5 time periods *)
u1[2] := u1[1]; u1[3] := u1[1]; u1[4] := u1[1]; u1[5] := u1[1];

(* And now we solve for the input magnitudes *)
Solve[{x1T == xf1},{u1[1]}] >>> pc_inputs.m

(* We know that xiT are linear in the inputs u2[i], namely
* final = Matrix.input2 + constant
* so we set goal = final and solve for the magnitudes of input2:
* solution = Inverse[Matrix].(goal - constant)
*)
input2 = {u2[1], u2[2], u2[3], u2[4], u2[5]};
final = { x2T, x3T, x4T, x5T, x6T };
goal = { xf2, xf3, xf4, xf5, xf6 };
Matrix = Jac[final,input2];
constant = final - Matrix.input2;
solution = (Inverse[Matrix].(goal - constant)) >>> pc_inputs.m
```

```
(* polynomial.m     D. Tilbury     January 14, 1992
* Here we take a 6-state system in single-chained, 2-input form
* and steer it with Piecewise constant controls over 5 steps,
* We do two cases:  u1 = 1, u1 = -1;
*)

(* Set up the initial and final conditions *)
x0 = {x01,x02,x03,x04,x05,x06};
xf = {xf1,xf2,xf3,xf4,xf5,xf6};

(* Choose the inputs to be polynomials *)
(* u1[t_] := 1 *)
u1[t_] := -1;
u2[t_] := b0 + b1 t + b2 t^2 + b3 t^3 + b4 t^4;

(* Calculate the states as functions of time *)
x1[t_] := x0[[1]] + Integrate[u1[tau],tau,0,t];
x2[t_] := x0[[2]] + Integrate[u2[tau],tau,0,t];
x3[t_] := x0[[3]] + Integrate[u1[tau] x2[tau],tau,0,t]
x4[t_] := x0[[4]] + Integrate[u1[tau] x3[tau],tau,0,t]
x5[t_] := x0[[5]] + Integrate[u1[tau] x4[tau],tau,0,t]
x6[t_] := x0[[6]] + Integrate[u1[tau] x5[tau],tau,0,t]

(* The time needed to steer is determined by the first state *)
Solve[x1[T] == xf1, T] >>> poly_inputs.m

(* Now we can solve for the desired coefficients.
* We know that final = Matrix.coeffs + const
* Therefore, setting goal = final, we find
* coeffs = Inverse[Matrix].(goal - const)
*)
coeffs = { b0, b1, b2, b3, b4 };
final = {x2[T],x3[T],x4[T],x5[T],x6[T]};
goal = { xf2, xf3, xf4, xf5, xf6 };
Matrix = Jac[final,coeffs];
const = final - Matrix.coeffs;
solution = Inverse[Matrix].(goal - const) >>> poly_inputs.m
```

REFERENCES

1. J. C. Alexander and J. H. Maddocks. On the maneuvering of vehicles. *SIAM Journal of Applied Mathematics*, 48(1):38–51, 1988.
2. J. Barraquand and J-C. Latombe. On nonholonomic mobile robots and optimal maneuvering. In *4th International Symposium on Intelligent Control*, pages 74–83, Albany, NY, 1989.
3. R. W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer-Verlag, New York, 1981.
4. R. L Bryant, S. S. Chern, R. B. Gardner, H. L. Goldschmidt, and P. A. Griffiths. *Exterior Differential Systems*. Springer-Verlag, 1991.
5. L. Bushnell, D. Tilbury, and S. S. Sastry. Steering chained form nonholonomic systems using sinusoids: The firetruck example. Technical Report UCB/ERL M92/107, University of California, Berkeley, Electronics Research Lab, 1992. To appear at the 1993 European Control Conference.
6. J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, 1988.
7. R. Chatila. Mobile robot navigation: Space modeling and decisional processes. In O. Faugeras and G. Giralt, editors, *Robotics Research: The Third International Symposium*, pages 373–378. MIT Press, 1986.
8. A. W. Divelbiss and J. Wen. A global approach to nonholonomic motion planning. In *Proceedings of the IEEE Control and Decision Conference*, pages 1597–1602, Tucson, Arizona, 1992.
9. C. Fernandes, L. Gurvits, and Z. Li. Foundations of nonholonomic motion planning. Technical Report Robotics Research Laboratory, Courant Institute of Mathematical Sciences, 1991.
10. C. Fernandes, L. Gurvits, and Z. Li. Optimal nonholonomic motion planning for a falling cat. In Z. Li and J. Canny, editors, *Nonholonomic Motion Planning*, pages 379–421. Kluwer Academic Publishers, 1993.
11. M. Fliess, J. Lévine, P. Martin, and P. Rouchon. On differentially flat nonlinear systems. In *Proceedings of the IFAC Nonlinear Control Systems Design Symposium*, pages 408–412, Bordeaux, France, 1992.
12. R. B. Gardner and W. F. Shadwick. The GS algorithm for exact linearization to Brunovsky normal form. *IEEE Transactions on Circuits and Systems*, 37(2):224–230, 1992.
13. G. Giralt, R. Chatila, and M. Vaisset. An integrated navigation and motion control system for autonomous multisensory mobile robots. In M. Brady and R. Paul, editors, *Robotics Research : The First International Symposium*, pages 191–214. MIT Press, Cambridge, Massachusetts, 1984.
14. L. Gurvits and Z. Li. Smooth time-periodic feedback solutions for nonholonmic motion planning. In Z. Li and J. Canny, editors, *Nonholonomic Motion Planning*, pages 53–108. Kluwer Academic Publishers, 1993.
15. J. Hopcroft J. T. Schwartz, M. Sharir. *Planning, Geometry and Complexity of Robot Motion*. Ablex, Norwood, New Jersey, 1987.
16. G. Jacob. Motion planning by piecewise constant or polynomial inputs. In *Proceedings of the IFAC Nonlinear Control Systems Design Symposium*, pages 628–633, Bordeaux, France, 1992.
17. G. Lafferriere and H. J. Sussmann. Motion planning for controllable systems without drift. In *Proceedings of the IEEE International Conference on Robotics and Automation*,

pages 1148–1153, Sacramento, California, 1991.
18. J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
19. J-P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. In L. O. Hertzberger and F. C. A. Green, editors, *Intelligent Autonomous Systems*, pages 346–354. North Holland, 1987.
20. J-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1120–1123, 1987.
21. J-P. Laumond. Nonholonomic motion planning versus controllability via the multibody car system example. Technical Report STAN-CS-90-1345, Stanford University, Department of Computer Science, December 1990.
22. J-P. Laumond. Controllability of a multibody mobile robot. In *Proceedings of the International Conference on Advanced Robotics*, pages 1033–1038, Pisa, Italy, 1991.
23. J-P. Laumond, P. Jacobs, M. Taïx, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 1993. In press.
24. Z. Li and J. Canny, editors. *Nonholonomic Motion Planning*. Kluwer Academic Publishers, 1993.
25. B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2533–2540, Nice, France, 1992.
26. S. Monaco and D. Normand-Cyrot. An introduction to motion planning under multirate digital control. In *Proceedings of the IEEE Control and Decision Conference*, pages 1780–1785, Tucson, Arizona, 1992.
27. R. M. Murray. Nilpotent bases for a class of non-integrable distributions with applications to trajectory generation for nonholonomic systems. Technical Report CIT/CDS 92-002, California Institute of Technology, 1992.
28. R. M. Murray and S. S. Sastry. Grasping and manipulation using multifingered robot hands. In R. W. Brockett, editor, *Robotics: Proceedings of Symposia in Applied Mathematics, Volume 41*, pages 91–128. American Mathematical Society, 1990.
29. R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. Technical Report UCB/ERL M91/45, Electronics Research Laboratory, University of California at Berkeley, 1991. To appear in *IEEE Transactions on Automatic Control*, May 1993.
30. J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.
31. C. Samson. Velocity and torque feedback control of a nonholonomic cart. In *International Workshop in Adaptive and Nonlinear Control: Issues in Robotics*, 1990.
32. O. J. Sørdalen. Conversion of the kinematics of a car with n trailers into a chained form. Submitted to the 1993 *IEEE International Conference on Robotics and Automation*, preprint.
33. H. J. Sussmann and W. Liu. Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories. In *Proceedings of the IEEE Control and Decision Conference*, pages 437–442, 1991.
34. A. Teel, R. M. Murray, and G. Walsh. Nonholonomic control systems: From steering to stabilization with sinusoids. In *Proceedings of the IEEE Control and Decision Conference*, pages 1603–1609, 1992.
35. D. Tilbury, J-P. Laumond, R. Murray, S. Sastry, and G. Walsh. Steering car-like systems

with trailers using sinusoids. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1993–1998, Nice, France, 1992.

36. G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J-P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1999–2004, Nice, France, 1992. To appear in *IEEE Transactions on Automatic Control*, November 1993.
37. Stephen Wolfram. *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley, Redwood City, California, second edition, 1991.