MOTION PLANNING FOR CONTROLLABLE SYSTEMS WITHOUT DRIFT

G. Lafferriere[1]
Department of Mathematical Sciences
P.O. Box 751, Portland State University
Portland, OR 97207-0751

hmgl@psuorpn.cc.pdx.edu

H. J. Sussmann[2]
Department of Mathematics
Rutgers University
New Brunswick, NJ 08903
Fax: 201-932-5530
sussmann@hamilton.rutgers.edu

Keywords: Motion planning, Nonholonomic constraints, Lie brackets

ABSTRACT

We propose a general strategy for solving the motion planning problem for real analytic, controllable systems without drift. The procedure starts by computing a control that steers the given initial point to the desired target point for an extended system, in which a number of Lie brackets of the system vector fields are added to the right-hand side. The main point then is to use formal calculations based on the product expansion relative to a P. Hall basis, to produce another control that achieves the desired result on the formal level. It then turns out that this control provides an exact solution of the original problem if the given system is nilpotent. When the system is not nilpotent, one can still produce an iterative algorithm that converges very fast to a solution. Using the theory of feedback nilpotentization, one can find classes of non-nilpotent systems for which the algorithm, in cascade with a precompensator, produces an exact solution in a finite number of steps. We also describe results of simulations which illustrate the effectiveness of the procedure.

§1. Introduction .

The purpose of this note is to present a general strategy for solving the Motion Planning Problem (MPP) for controllable systems without drift. Following the line of research initiated by Brockett and Sastry (cf. e.g. [2], [9]), we will propose a strategy based on tools from "differential geometric control theory," and making systematic use of Lie brackets of vector fields and Lie algebraic properties.

We consider control systems:

$$\Sigma : \qquad \dot{x} = u_1 f_1(x) + \ldots + u_m f_m(x) \ , \tag{1.1}$$

where

1. $f_1, \ldots, f_m$ are *real analytic* vector fields on $\mathbb{R}^n$,

2. Σ is completely controllable.

Notice that the right-hand side of (1.1) does not contain a term of the form $f_0(x)$, that is, we are assuming that *there is no drift.*

It is well known that Condition **2** is equivalent to the LARC (Lie algebra rank condition), i.e. to the condition that $L(\mathbf{f})(x) = \mathbb{R}^n$ for all x. Here

1. $L(\mathbf{f})$ is the *controllability Lie algebra* of Σ, i.e. the Lie algebra of vector fields generated by $\mathbf{f} = \{f_1, \ldots, f_m\}$.

and

2. If S is any set of vector fields, we write:

$$S(x) \stackrel{\text{def}}{=} \{X(x) : X \in S\} . \tag{1.2}$$

The MPP (Motion Planning Problem) is the problem of finding reasonable algorithms producing, for every pair p, q of points, an open-loop control

$$t \to u(t) = (u_1(t), \ldots, u_m(t)) \tag{1.3}$$

that steers p to q.

The controllability condition LARC guarantees that such a control exists for every p and q. However, here we are not satisfied with just knowing that a control exists. We want a reasonable procedure that will compute one. Since "reasonable procedure" is not a uniquely defined mathematical concept, the problem, as stated, does not have a unique solution. Whether or not a particular proposed solution is satisfactory is to a large extent a matter of judgement, and different solutions may be preferable in particular situations for different reasons.

In this report we describe one solution, outline the theory behind it, and present some data from simulations.

The MPP has been studied by a number of authors, notably Sastry, Hauser, Murray, Li et al., cf. [4], [8], [9], [11]. They have proposed procedures that work very well in a number of special cases. Often, these procedures

(a) attempt to use optimal control;

(b) require special assumptions (e.g. that for every k the span of the Lie brackets up to order k has constant dimension);

(c) work in special cases.

The strategy proposed here is a modification of the above-mentioned approach, in that

(a) it does not in principle require special assumptions on the spans of the Lie brackets,

(b) it does not use optimal control,

(c) it works exactly for *nilpotent* and *nilpotentizable systems* (defined below),

(d) it works approximately for completely general systems, and can be used to produce a "successive approximations" algorithm that converges quite fast to an exact solution.

§2. Nilpotent Systems .

We recall that a Lie algebra L is said to be *nilpotent* if there is an integer $k > 0$ with the property that all the Lie brackets $[v_1, [v_2, \ldots, [v_k, v_{k+1}] \ldots]]$ vanish. The smallest such k is the *order of nilpotency* of L, and L is said to be *nilpotent of order k*.

The system Σ will be called *nilpotent* if its controllability Lie algebra $L(\mathbf{f})$ is a nilpotent Lie algebra.

Example 2.1. The system Σ is nilpotent of order one if $[f_i, f_j] = 0$ for all i, j, i.e. if $L(\mathbf{f})$ is abelian.

If $[f_i,[f_j,f_k]]=0$ for all i, j, k, but $[f_i,f_j]\neq 0$ for some i, j, then Σ is nilpotent of order 2.

§3. The Strategy .

The main point of the strategy proposed here is to consider, in conjunction with the original system Σ, an *extended system*

$$\Sigma_e: \qquad \dot{x}=v_1f_1(x)+\ldots+v_mf_m(x)+v_{m+1}f_{m+1}(x)+\ldots+v_rf_r(x)\ , \tag{3.1}$$

where $f_{m+1},\ldots,f_r$ are higher-order Lie brackets of the f_i, chosen so that $f_1(x),\ldots,f_r(x)$ span $\mathbb{R}^n$ for all x, or at least for all x in some prescribed bounded region $\mathcal{R}$.

For instance, if $n=3$, $m=2$, and f_1, f_2 are such that $f_1(x),f_2(x),[f_1,f_2](x)$ are linearly independent at each point, then we could take f_3 to be $[f_1,f_2]$, and we could let our extended system be

$$\Sigma_e: \qquad \dot{x}=v_1f_1(x)+v_2f_2(x)+v_3f_3(x)\ . \tag{3.2}$$

However, this is not the only possible choice. We could also use an even larger extension, if desired, e.g. by taking $f_4=[f_1,[f_1,f_2]]$, and letting $r=4$.

Remark 3.1. We prefer to allow r to be larger than n, because

a. as explained below in §4, the computation of v requires the determination of the inverse of the matrix $F(x)=\mathrm{col}\,(f_1(x),\ldots,f_r(x))$ —if $r=n$— or of a pseudoinverse —if $r>n$. By adding more vector fields than the bare minimum number n needed to span $\mathbb{R}^n$, we make it easier for F to have an $n\times n$ submatrix with a better condition number.

b. the LARC implies that for every point $\bar{x}$ there is a set of n brackets that spans $\mathbb{R}^n$ at all points in some neighborhood U of $\bar{x}$. However, one usually wants to work on some fixed large bounded region $\mathcal{R}$. In general, for such a region there will be a fixed finite set of brackets that spans at every point, but it will not be possible to take it to consist of exactly n brackets. ∎

Once we have selected an extended system Σ_e, the solution proposed here of the MPP involves two basic steps:

STEP I: Find a control v that steers p to q for Σ_e.

STEP II: Use v to compute a control u that steers p to q for Σ.

As explained below, Step I is in principle easy. Step II is harder and involves some interesting algebra, that will be discussed in detail below.

§4. The First Step .

To find v, we begin by choosing a C^1 curve $\gamma:[0,T]\to\mathbb{R}^n$ that goes from p to q. This, of course, is quite easy to do. For instance, a straight-line segment will work perfectly well. However, in some situations it may be desirable to start with some other curve, e.g. if we are dealing with an obstacle avoidance problem.

Having chosen γ, we then write the tangent vector $\dot{\gamma}(t)$ as a linear combination of the vectors $f_1(\gamma(t)),\ldots,f_r(\gamma(t))$. The resulting coefficients are the $v_i(t)$. The problem of expressing $\dot{\gamma}(t)$ as a linear combination of $f_1(x),\ldots,f_r(x)$ involves inverting a square matrix (if $r=n$) or computing a pseudoinverse.

§5. Outline of Step II .

We now begin our discussion of Step II. We will first give a rough outline, and then proceed to fill in the details.

It will be important to restrict our choice of the extended system a bit further, by requiring the "new" f_i to be brackets arising from a *P. Hall basis.* (The meaning of this is explained below.)

Step II then proceeds as follows. We choose k such that all the $f_i, i>m$ have degree $\leq k$. Then:

II. We do *formal calculations* on the free nilpotent Lie algebra $L_k(X_1,\ldots,X_m)$. The X_i are purely formal *noncommuting indeterminates.* The calculations are done in two steps:

IIa: Solve a simple differential equation (the "formal extended equation") on the nilpotent Lie group $G_k(X_1,\ldots,X_m)$ (cf. §7), using P. Hall coordinates.

IIb: Find u from the P. Hall coordinates obtained in IIa.

§6. P. Hall Bases .

We begin by recalling the definition of the free associative and Lie algebras in a collection of indeterminates. (For more details, cf., e.g., [12] and [14].)

We use $A(X_1,\ldots,X_m)$ to denote the algebra of noncommutative polynomials in $(X_1,\ldots,X_m)$. For example, if $m=2$ and $X_1=X$, $X_2=Y$, then a basis of $A(X_1,\ldots,X_m)$ is given by 1, X, Y, XX, XY, YX, YY, XXX, XXY, XYX, ...

Like all associative algebras, $A(X_1,\ldots,X_m)$ is also a Lie algebra, with the Lie bracket defined by $[P,Q]=PQ-QP$. We then define $L(X_1,\ldots,X_m)$ to be the Lie subalgebra of $A(X_1,\ldots,X_m)$ generated by $X_1,\ldots,X_m$. The elements of $L(X_1,\ldots,X_m)$ are known as *Lie polynomials* in $X_1,\ldots,X_m$.

We also use $\hat{A}(X_1,\ldots,X_m)$ and $\hat{L}(X_1,\ldots,X_m)$ to denote, respectively, the set of *noncommutative formal power series* and that of *Lie series* in the X_i. That is, $\hat{A}(X_1,\ldots,X_m)$ consists of all (not necessarily finite) linear combinations $\sum_M c_M M$ of monomials, and $\hat{L}(X_1,\ldots,X_m)$ is the set of those $S\in\hat{A}(X_1,\ldots,X_m)$ such that, for each j, the j-th homogeneous component of S is in $L(X_1,\ldots,X_m)$.

The nilpotent versions $A_k(X_1,\ldots,X_m)$, $L_k(X_1,\ldots,X_m)$ (also denoted A_k^m, L_k^m) are then defined by just killing all the monomials of degree $k+1$. (In our example, a basis of $A_2(X_1,\ldots,X_m)$ is given by 1, X, Y, XX, XY, YX, YY, and the multiplication table is such that, e.g., $X(XX)=X(XY)=\ldots=0$. A basis of $L_2(X_1,\ldots,X_m)$ is given by X, Y, and $[X,Y]$.) Then A_k^m and L_k^m are known, respectively, as the *free nilpotent associative algebra of order k* and the *free nilpotent Lie algebra of order k*.

Writing down a basis of A_k^m is always very easy, for all one has to do is list all the possible monomials of degree $\le k$. Writing down a basis of L_k^m is harder. For instance, a basis of $L_3(X,Y)$ is: $X,Y,[X,Y],[X,[X,Y]],[Y,[X,Y]]$. Notice that we did not include $[Y,X]$ or $[X,[Y,X]]$, i.e. we did not take our basis to consist of all "Lie monomials." In general, to get a basis of $L_k(X_1,\ldots,X_m)$ one should *not* use all the formal brackets of degree $\le k$. The reason is that among these brackets there are *relations* (skew-symmetry, Jacobi identity).

The concept of a *P. Hall basis* is a particular way to get around this difficulty and to select a basis. Precisely, a P. Hall basis of $L(X_1,\ldots,X_m)$ is defined as follows.

We first let $\mathcal{F}_m$ be the set of *formal Lie monomials* in $X_1,\ldots,X_m$. The elements of $\mathcal{F}_m$ are words in the alphabet whose elements are the indeterminates $X_1,\ldots,X_m$, the comma, and the left and right square brackets. The set $\mathcal{F}_m$ is constructed recursively by requiring that (1) each X_i be in $\mathcal{F}_m$, (2) if A and B are in $\mathcal{F}_m$ then $[A,B]\in\mathcal{F}_m$, (3) $\mathcal{F}_m$ is the smallest set of words that satisfies (1) and (2).

Notice that the elements of $\mathcal{F}_m$ are to be regarded as purely formal expressions, *not* as elements of $L(X_1,\ldots,X_m)$. For instance, $[X,[X,Y]]$ and $[[Y,X],X]$ are *different* as elements of $\mathcal{F}_m$, although of course they are equal in $L(X,Y)$. Notice also that every element of $\mathcal{F}_m$ which is not one of the indeterminates X_i can be written in a unique way as a bracket $[A,B]$, with $A,B\in\mathcal{F}_m$. (This is not true of elements of $L(X_1,\ldots,X_m)$. For instance, if $P=[X,[X,Y]]$, then we can take $A=X$ and $B=[X,Y]$. But P is also equal to $[[Y,X],X]$, so we can also take $A=[Y,X]$, $B=X$.)

A *P. Hall basis* of $L(X_1,\ldots,X_m)$ is then a totally ordered subset $(\mathcal{B},\prec)$ of $\mathcal{F}_m$, such that

PH1. The X_i belong to $\mathcal{B}$.

PH2. If $A,B\in\mathcal{B}$, and degree$(A)<$ degree(B), then $A\prec B$.

PH3. If $P\in\mathcal{F}_m$, and P is not one of the X_i, then $P\in\mathcal{B}$ if and only if $P=[A,B]$ with $A,B\in\mathcal{B}$, $A\prec B$, and either $B=X_i$ for some i or $B=[C,D]$ with $C,D\in\mathcal{B}$, $C\preceq A$.

Example 6.1. Let $m = 2$ as before, and let the indeterminates be X, Y. Then, up to degree five, the following constitute a P. Hall basis: X, Y, $[X,Y]$, $[X,[X,Y]]$, $[Y,[X,Y]]$, $[X,[X,[X,Y]]]$, $[Y,[X,[X,Y]]]$, $[Y,[Y,[X,Y]]]$, $[X,[X,[X,[X,Y]]]]$, $[Y,[X,[X,[X,Y]]]]$, $[Y,[Y,[X,[X,Y]]]]$, $[Y,[Y,[Y,[X,Y]]]]$, $[[X,Y],[X,[X,Y]]]$, and $[[X,Y],[Y,[X,Y]]]$. ∎

The main fact about P. Hall bases is that, as the name indicates, they solve the problem of picking a basis of $L(X_1,\ldots,X_m)$. Precisely, if $\mathcal{E}$ is any subset of $\mathcal{F}_m$, we let $\hat{\mathcal{E}}$ denote the set of all elements of $L(X_1,\ldots,X_m)$ obtained by actually evaluating the Lie brackets. We then have:

1. if $(\mathcal{B},\prec)$ is a P. Hall basis of $L(X_1,\ldots,X_m)$, then $\hat{\mathcal{B}}$ is a basis of $L(X_1,\ldots,X_m)$.
2. the set $\{M \in \hat{\mathcal{B}} : \text{degree}(M) \le k\}$ is a basis of L_k^m.

Any set $\{M \in \hat{\mathcal{B}} : \text{degree}(M) \le k\}$, where $(\mathcal{B},\prec)$ is a P. Hall basis of $L(X_1,\ldots,X_m)$, will be called a *P. Hall basis of* L_k^m.

§7. P. Hall Coordinates .

It is clear that, if P is any element of $\hat{A}(X_1,\ldots,X_m)$ or $A_k(X_1,\ldots,X_m)$ that has no constant term, then the exponential e^P is well defined by means of the usual power series.

We define

$$\begin{aligned} \hat{G}(X_1,\ldots,X_m) &= \{e^Z : Z \in \hat{L}_k(X_1,\ldots,X_m)\} , \\ G_k^m = G_k(X_1,\ldots,X_m) &= \{e^Z : Z \in L_k(X_1,\ldots,X_m)\} . \end{aligned} \tag{7.1}$$

(Examples of elements of $\hat{G}(X,Y)$: $e^X = 1 + X + \frac{1}{2}X^2 + 1/6X^3 + \ldots$, $e^{[X,Y]} = 1 + [X,Y] + \frac{1}{2}[X,Y]^2 + \ldots$, $e^{X-Y+3[X,Y]} = 1 + X - Y + 3[X,Y] + \ldots$. Examples of elements of $G_2(X,Y)$: $e^X = 1 + X + \frac{1}{2}X^2$, $e^{[X,Y]} = 1 + [X,Y]$, $e^{X-Y+3[X,Y]} = 1 + X - Y + 3[X,Y] + \frac{1}{2}(X-Y)^2$. Notice that the meaning of e^P depends on whether P is regarded as an element of $\hat{G}(X_1,\ldots,X_m)$ or of $G_k(X_1,\ldots,X_m)$.) Then both $\hat{G}(X_1,\ldots,X_m)$ and G_k^m are closed under multiplication, thanks to the Campbell-Hausdorff Formula. Moreover, since $e^{-Z}e^Z = 1$, they are both groups. It is easy to see that G_k^m is the connected simply connected Lie group with Lie algebra L_k^m. We call G_k^m the *free nilpotent Lie group of order k with m infinitesimal generators.*

We now let B_1, B_2, …, B_s be a P. Hall basis of $L_k(X_1,\ldots,X_m)$. Then it is well known that every $S \in G_k(X_1,\ldots,X_m)$ has unique expressions

$$S = e^{h_s B_s} e^{h_{s-1} B_{s-1}} \ldots e^{h_2 B_2} e^{h_1 B_1} \ , \tag{7.2}$$

$$S = e^{\tilde{h}_1 B_1} e^{\tilde{h}_2 B_2} \ldots e^{\tilde{h}_{s-1} B_{s-1}} e^{\tilde{h}_s B_s} \ . \tag{7.3}$$

The maps $S \to (h_1,\ldots,h_s)$ and $S \to (\tilde{h}_1,\ldots,\tilde{h}_s)$ are global coordinate charts on G_k^m, and establish global diffeomorphisms between G_k^m and $\mathbb{R}^s$. The h_i are the *backward P. Hall coordinates* of S. The $\tilde{h}_i$ are the *forward P. Hall coordinates* of S.

§8. The formal extended equation .

We are now ready to give more details about Step IIa. We fix a P. Hall basis of the Lie algebra $L_k(X_1,\ldots,X_m)$, where k is chosen as explained in §5. We let $\mathbf{E_f}$ be the "evaluation map" that assigns to each $P \in L(X_1,\ldots,X_m)$ the vector field obtained by plugging in the f_i, $i=1,\ldots,m$, for the corresponding X_i. (For example, $\mathbf{E_f}([X_3,[X_1,[X_2,X_3]]]) = [f_3,[f_1,[f_2,f_3]]]$.) We assume that the vector fields $f_{m+1},\ldots,f_r$ are given by $f_j = \mathbf{E_f}(X_j)$ for $j = m+1,\ldots,r$, where $X_{m+1},\ldots,X_r$ are such that $X_1,\ldots,X_r$ is a P. Hall basis of L_k^m.

We then solve the O.D.E.

$$\Sigma_{fe}: \qquad \begin{aligned} \dot S(t) &= S(t)\Big(v_1(t)X_1 + \ldots + v_r(t)X_r\Big) \ , \\ S(0) &= 1 \ , \end{aligned} \tag{8.1}$$

in backward P. Hall coordinates. (This O.D.E. will be called the "formal extended equation.") That is, we write the solution as a product

$$S(t) = \mathrm{e}^{h_s(t)B_s}\mathrm{e}^{h_{s-1}(t)B_{s-1}}\ldots\mathrm{e}^{h_2(t)B_2}\mathrm{e}^{h_1(t)B_1} \ . \tag{8.2}$$

The advantage of using backward P. Hall coordinates is that *the functions $h_j(t)$ are easily computed by solving a system of O.D.E.'s with input v.* This system is solved by just successive integrations and algebraic operations. The proof that the computation of the h_i can be done by integrations was given in [13] for the simpler case in which the system Σ_{fe} is not extended. An almost equally simple proof works for the extended case. (The details will be given in [15].) The following example illustrates what happens in general.

Example 8.1. Say $k=3$, $m=2$, $r=4$, $f_3=[f_1,f_2]$, $f_4=[f_1,[f_1,f_2]]$. Then we can take

$$\begin{aligned} B_1 &= X \ , \\ B_2 &= Y \ , \\ B_3 &= [X,Y] \ , \\ B_4 &= [X,[X,Y]] \ , \\ B_5 &= [Y,[X,Y]] \ . \end{aligned} \tag{8.3}$$

Then $X_3=[X,Y]$, $X_4=[X,[X,Y]]$, $X_5=[Y,[X,Y]]$. Since $r=4$, we solve (8.1) with $v_5 \equiv 0$. Then $h_1(t),\ldots,h_5(t)$ are computed by solving

$$\begin{aligned} \dot h_1 &= v_1 \ , \\ \dot h_2 &= v_2 \ , \\ \dot h_3 &= h_1v_2 + v_3 \ , \\ \dot h_4 &= \frac{1}{2}h_1^2v_2 + h_1v_3 + v_4 \ , \\ \dot h_5 &= h_2v_3 + h_1h_2v_2 \ , \\ h_1(0) = h_2(0) &= h_3(0) = h_4(0) = h_5(0) = 0 \ . \end{aligned}$$

Before we proceed to Step IIb, we introduce an intermediate step. In this step, we transform from backward to forward P. Hall coordinates. It is not hard to see that this amounts to a simple algebraic transformation. Alternatively, we could dispense with this intermediate step by slightly modifying Step IIa. Instead of solving the formal extended equation as above, we solve it using the input $t \to -v(T-t)$ rather than $t \to v(t)$. It is easily seen that the end result is $S(T)^{-1}$, where $S(T)$ is the result of solving the formal extended equation with the input v. Once we have $S(T)^{-1}$ in backward P. Hall coordinates, it is an easy matter to obtain $S(T)$ in forward P. Hall coordinates. All we need is to reverse the order of the exponentials and to change all the signs.

Remark 8.1. Equation (8.1) is formally analogous to the equation that defines the "Chen series," introduced in control theory by M. Fliess (cf. [3]), although it differs from it in two ways: (a) in the definition of the Chen series one does not have the extra brackets $X_{m+1}\ldots, X_r$, and (b) (8.1) is interpreted as evolving in A_k^m rather than in the algebra of formal power series. ∎

§9. Finding u .

We now proceed to Step IIb. In this step, we work with the "formal equation"

$$\begin{aligned} \dot{S}^*(t) &= S^*(t)\Big(u_1(t)X_1 + \ldots + u_m(t)X_m\Big) \ , \\ S^*(0) &= 1 \ , \end{aligned} \tag{9.1}$$

which is the same as the formal extended equation, except that it is *not* extended, i.e. only the indeterminates X_i, $i = 1, \ldots, m$, but no high-order brackets, appear in the right-hand side.

The goal of Step IIb is to solve the *inverse problem* of finding controls u_i, $i = 1, \ldots, m$ that, when plugged into (9.1), give rise to an $S^*(T)$ which is equal to $S(T)$.

That is, we have to solve the following problem:

P *given P. Hall coordinates $h_1, h_2, \ldots, h_s$ of an $S \in G_k(X_1, \ldots, X_m)$, find a control $t \to u(t)$ that gives rise to an $S^*(T)$ that has these coordinates.*

Problem **P** can be solved in a number of ways. Here we describe one of several possible solutions. The main idea is to solve separately for each exponential factor, and then concatenate the results. It turns out that this is quite easy to do, if we only ask to do it "modulo factors of higher order." For example, as we will show below, if we want to find a u for a given factor $e^{\alpha[X,Y]}$, the solution is quite simple, provided that we are willing to settle for a little bit less, namely, for something of the form

$$e^{\alpha[X,Y]}e^{\beta[X,[X,Y]]}e^{\gamma[Y,[X,Y]]} \cdots .$$

It then turns out that the extra factors can be combined with those high-order factors that are already present, yielding a new product of high order factors. If we continue in this way, then the nilpotency implies that the procedure eventually stops, and when it does an exact solution u is obtained.

We illustrate this with an example.

Example 9.1. *Find u for*

$$S(T) = e^{\alpha X}e^{\beta Y}e^{\gamma[X,Y]}e^{\delta[X,[X,Y]]}e^{\varepsilon[Y,[X,Y]]} \ . \tag{9.2}$$

To solve this, fix once and for all open-loop controls A, B that give rise to e^X, e^Y. Use xA for "x times A," and so on. Use # for concatenation. (So $A\#B$ means 'A followed by B.')
Then:

- $\alpha A\#\beta B$ gives rise to $e^{\alpha X}e^{\beta Y}$
- $C = \sqrt{\gamma}A\#\sqrt{\gamma}B\#(-\sqrt{\gamma}A)\#(-\sqrt{\gamma}B)$ "almost" gives rise to $e^{\gamma[X,Y]}$. (We are assuming that $\gamma > 0$. If $\gamma < 0$ we just interchange X and Y.) Precisely, C gives rise to

$$e^{\gamma[X,Y]}e^{\tilde{\gamma}[X[X,,Y]]}e^{-\tilde{\gamma}[Y,[X,Y]]}\ , \tag{9.3}$$

where $\tilde{\gamma} = \frac{1}{2}\gamma^{\frac{3}{2}}$.

So far, we have shown that

$$\alpha A\#\beta Y\#\sqrt{\gamma}A\#\sqrt{\gamma}B\#(-\sqrt{\gamma}A)\#(-\sqrt{\gamma}B)$$

gives rise to

$$e^{\alpha X}e^{\beta Y}e^{\gamma[X,Y]}e^{\tilde{\gamma}[X,[X,Y]]}e^{-\tilde{\gamma}[Y,[X,Y]]}\ .$$

What we really want is

$$e^{\alpha X}e^{\beta Y}e^{\gamma[X,Y]}e^{\delta[X,[X,Y]]}e^{\varepsilon[Y,[X,Y]]}\ .$$

To achieve this, it suffices to find a control that gives rise to $e^{\tilde{\delta}[X,[X,Y]]}e^{\tilde{\varepsilon}[Y,[X,Y]]}$, where $\tilde{\delta} = \delta - \tilde{\gamma}$, $\tilde{\varepsilon} = \varepsilon + \tilde{\gamma}$. For this we could use

$$\begin{gathered}\rho A\#\rho A\#\rho B\#(-\rho A)\#(-\rho B)\#(-\rho A)\#\rho B\#\rho A\#(-\rho B)\#(-\rho A)\# \\ \sigma B\#\sigma A\#\sigma B\#(-\sigma A)\#(-\sigma B)\#(-\sigma B)\#\sigma B\#\sigma A\#(-\sigma B)\#(-\sigma A)\ ,\end{gathered}$$

where $\rho = (\tilde{\delta})^{\frac{1}{3}}$, $\sigma = (\tilde{\varepsilon})^{\frac{1}{3}}$.

The end result is a control u made of 26 pieces that solves our problem exactly. In other words, $S(T)$ is realized in 26 moves. ∎

§10. The main theorems .

So far, we have described a procedure for computing a control u, but have said nothing about what happens when we plug u into our original equation (1.1).

The answer to this is provided by the following two theorems, an outline of whose proof is given in [7]. (A detailed proof will be given in [15]. The first theorem is essentially a consequence of the results of Palais [10] on integrability of finite-dimensional actions of Lie algebras.) Recall that a vector field is *complete* if its integral curves are defined for all times.

Theorem 10.1. *If $L(\mathbf{f})$ is nilpotent of order k and the vector fields $f_1, \ldots, f_m$ are complete, then the u computed by the above method steers p to q exactly.* ∎

Remark 10.1. If the f_i are not complete then it may happen that the trajectory for a given control u is not defined for all t. Moreover, even when the trajectory is defined the control may fail to steer p to q. ∎

Theorem 10.2. *For a general system (1.1), let $\sigma(p,q,u)$ be the point to which u steers p. Then, on any bounded region $\mathcal{R}$, the error $E(p,q) = ||\sigma(p,q,u) - q||$ satisfies a bound*

$$E(p,q) \leq F(\Delta)\Delta^{\theta}\ , \tag{10.1}$$

for all $p, q \in \mathcal{R}$, where

$$\Delta = ||p - q||\ , \tag{10.2}$$

$$\theta = 1 + \frac{1}{k}\ , \tag{10.3}$$

and $F : [0,\infty) \to [0,\infty]$ is a function which is finite and bounded for Δ near 0.

Remark 10.2. The function F is allowed to be infinite if Δ is sufficiently large. This is because in Theorem 10.2 we are not assuming completeness, so it may in fact happen that the trajectory for u is not even defined for all t.

Corollary 10.1. *If $\mathcal{R}$ is a bounded region, then there exists $\Delta > 0$ such that, if $p, q \in \mathcal{R}$ and $||p - q|| \leq \Delta$, then $E(p,q) \leq \frac{1}{2}||p - q||$.* ∎

Definition 10.1. The supremum of all the numbers Δ that satisfy the condition of Corollary 10.1 is called the *critical distance* for $\mathcal{R}$, and is denoted by $\Delta_c(\mathcal{R})$. Any number $\bar{\Delta}$ such that $0 < \bar{\Delta} \leq \Delta_c(\mathcal{R})$ will be called an *admissible step length* for $\mathcal{R}$.

§11. The iterated algorithm .

So far, we have explained how to solve the MPP if the system is exactly nilpotent. For a general system, the control u obtained by the prescription of the preceding sections is only an approximate solution. It turns out, however, that we can use our method to obtain a control that steers p to q within any arbitrary prescribed error. This is done by *iterating* the algorithm, and using always distances not greater than the critical distance $\Delta_c(\mathcal{R})$.

The iterated algorithm works as follows. We work in a fixed bounded region $\mathcal{R}$, and fix once and for all an admissible step length $\bar{\Delta}$ for $\mathcal{R}$. Suppose we are given p, q, and an upper bound ε for the permissible error. Then:

1. set $p_0 = p$, $i = 0$, and go to **2**.

2. Apply the algorithm to go from p_i to q_i, where

$$q_i = p_i + \min\Big(1, \frac{\bar{\Delta}}{||q - p_i||}\Big)(q - p_i)\,. \tag{11.1}$$

Let p_{i+1} be the resulting point.

3. If $||p_{i+1} - q|| \leq \varepsilon$ stop. Else $i := i + 1$ and go to **2**.

We refer to the above as the *iterated algorithm.*

Theorem 11.1. *Let $\mathcal{R}$ be a bounded region. Let Δ_c be the critical distance for $\mathcal{R}$, and let $\bar{\Delta}$ be such that $0 < \bar{\Delta} \leq \Delta_c$. Assume $||p - q|| = \nu\bar{\Delta}$. Then the iterated algorithm stops in at most $2[\nu] + \log_2(\frac{\bar{\Delta}}{\varepsilon})$ steps.*

The proof of this theorem consists of a simple geometric argument. Suppose that we are at the beginning of Step i, i.e. we are about to start **2** with some i and the corresponding p_i. Suppose $||p_i - q|| > \bar{\Delta}$. Then the algorithm will try to steer p_i to q_i, where q_i lies in the segment from p_i to q and is at a distance $\bar{\Delta}$ from p_i and at a distance $||p_i - q|| - \bar{\Delta}$ from q. However, the control u computed by the algorithm will actually steer p_i to p_{i+1}, where p_{i+1} satisfies $||p_{i+1} - q_i|| \leq \frac{1}{2}\bar{\Delta}$. By the triangle inequality, $||p_{i+1} - q|| \leq ||p_i - q|| - \frac{1}{2}\bar{\Delta}$. So, as long as $||p_i - q|| > \bar{\Delta}$, in each step we get closer to q by at least $\frac{1}{2}\bar{\Delta}$. It follows that, in at most $2[\nu]$ steps, we will get to a point p_i whose distance to q is $\leq \bar{\Delta}$. Once this has been achieved, each step diminishes the distance by a factor of 2 (at least). The conclusion follows easily. ∎

§12. Feedback Nilpotentization .

We now describe a method that, in many cases, can be used to produce an exact solution in a finite number of steps, even when a system is not nilpotent. The key fact here is that *many systems that are not nilpotent can be made nilpotent by means of a feedback transformation.*

Recall that a *feedback transformation* of Σ consists of a linear change of controls

$$u_i = \sum_{j=1}^{m} \beta_{ij}(x)v_j \ , \tag{12.1}$$

such that $\beta(x)$ is a nonsingular matrix for each x, and $\beta(x)$ is smooth as a function of x.

Let us call a system Σ *feedback nilpotentizable* (FN) if it can be made nilpotent by a feedback transformation.

Feedback nilpotentization was studied by H. Hermes, A. Lundell and D. Sullivan in [5] and [6]. They proved, for instance, that any system

$$\dot{x} = u_1 f_1(x) + u_2 f_2(x)\ ,\ \ x \in \mathbb{R}^3 \tag{12.2}$$

for which $f_1(\bar{x}), f_2(\bar{x}), [f_1, f_2](\bar{x})$ are linearly independent is feedback nilpotentizable on a neighborhood of $\bar{x}$.

In many cases, the nilpotentization of a system is not only possible in principle, but can actually be carried out in practice with explicit formulas for the components of the matrix β. When this happens, the MPP can of course be solved exactly, by using first a precompensator to implement the nilpotentizing feedback transformation. Examples of this will be shown below.

§13. Results from simulations .

In this section we present examples which illustrate the proposed techniques. More details about these examples, including plots of the resulting controls and trajectories, can be found in [7].

We consider first the problem of steering a unicycle. The controls are the driving speed and the steering speed. The equations for this system are

$$\begin{aligned}\dot{x}_1 &= \cos(x_3)u_1\ ,\\ \dot{x}_2 &= \sin(x_3)u_1\ ,\\ \dot{x}_3 &= u_2\ .\end{aligned} \tag{13.1}$$

where (x_1, x_2) are the Cartesian coordinates of the center of the unicycle and x_3 is the angle its main axis makes with the x_1-axis. We can rewrite this system as $\dot{x} = u_1 f_1(x) + u_2 f_2(x)$ where

$$f_1(x) = (\cos(x_3), \sin(x_3), 0) \quad f_2(x) = (0, 0, 1)\ . \tag{13.2}$$

Clearly the vectors $f_1(x), f_2(x), [f_1, f_2](x)$ span $\mathbb{R}^3$ near $x = 0$, so (13.1) is locally nilpotentizable as mentioned earlier in §12. However, (13.1) is not nilpotent. In fact,

$$\mathrm{ad}_{f_2}^{2n}(f_1) = (-1)^n(\cos(x_3), \sin(x_3), 0) \qquad n \geq 0. \tag{13.3}$$

We generated first an approximate trajectory for the system using the nilpotent formal system of order 2. In this case $\mathrm{e}^{[X,Y]} = \mathrm{e}^X \mathrm{e}^Y \mathrm{e}^{-X} \mathrm{e}^{-Y}$, and the formal solution can be written as

$$S(T) = \mathrm{e}^{\alpha_5 X}\mathrm{e}^{\alpha_4 Y}\mathrm{e}^{\alpha_3 X}\mathrm{e}^{\alpha_2 Y}\mathrm{e}^{\alpha_1 X}\ . \tag{13.4}$$

We can realize the above product of exponentials by choosing a piecewise constant control made up of 5 pieces, each piece generating a move corresponding to a trajectory of either f_1 or f_2.

Specifically, we tried to drive the system from $x_0 = (0, 0, -0.6)$ to $x_f = (3, 3, 0)$. As reference trajectory to compute the v's we used the straight-line segment between the two points parametrized from 0 to 1. constant control with $|u_i| \leq 1$ and applied it for as long as necessary so that its integral equals the corresponding exponent in the formal product expansion. The backward P. Hall coordinates for our choice of v's are $(-3.2, 0.6, 1.95)$. (We are using an order 2 approximation.) This suggests, for example, a large motion (given by the first coordinate) in the direction of $[f_1, f_2]$ which near the origin is approximately $(0, -1, 0)$. However, higher order brackets contribute to changes in that direction in part offsetting the changes in the y coordinate and in part contributing to larger changes in the x coordinate. Though the resulting trajectory does not reach the target, it does bring the vehicle closer. After two more iterations the trajectory reaches the target within the desired error $\varepsilon \leq 0.01$.

This procedure corresponds to guessing that the critical distance is larger than $\|x_0 - x_f\|$. If, however, we start with a smaller guess Δ_g for the critical distance, then the proposed technique requires that we try to move in steps not longer than Δ_g. Naturally, this will result in the final trajectory staying closer to the segment from x_0 to x_f. It turns out that it takes 6 steps to reach the desired target. It is worth noting that if the critical distance were actually 1, Theorem 10.2 gives an upper bound of about 15 for the number of steps.

Since this system is nilpotentizable we can actually get to the target point in one step exactly. More precisely, this system can be made nilpotent using the following feedback

$$\begin{aligned} u_1 &= \frac{1}{\cos(x_3)} w_1 \ , \\ u_2 &= \cos^2(x_3) w_2 \ . \end{aligned} \tag{13.5}$$

The system then becomes

$$\begin{aligned} \dot{x}_1 &= w_1 \ , \\ \dot{x}_2 &= \tan(x_3) w_1 \ , \\ \dot{x}_3 &= \cos^2(x_3) w_2 \ . \end{aligned} \tag{13.6}$$

which is nilpotent of order 2. We therefore compute the desired controls as follows. First apply the procedure to the nilpotent system to obtain the controls w_i. Then using the feedback (13.5) compute the controls for the original system.

The results of such a simulation (with the same x_0, x_f as before) show that the desired objective is achieved perfectly (cf. [7]). The only errors are those due to numerical integration.

A second example is given by a front wheel drive cart.

The controls are the driving speed of the front wheels (u_1) and the turning speed of the front wheels (u_2). The system equations are

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)\cos(x_4) u_1 \ , \\ \dot{x}_2 &= \cos(x_3)\sin(x_4) u_1 \ , \\ \dot{x}_3 &= u_2 \ , \\ \dot{x}_4 &= \frac{1}{l}\sin(x_3) u_1 \ . \end{aligned} \tag{13.7}$$

where x_1, x_2 are the Cartesian coordinates of the cart, x_3 is the steering angle and x_4 is the angle the main axis of the cart makes with the x_1-axis. Let's rewrite the system as $\dot{x} = u_1 f_1(x) + u_2 f_2(x)$. Then the vectors $f_1(x), f_2(x), [f_1, f_2](x), [f_1, [f_1, f_2]](x)$ span $\mathbb{R}^4$ in a neighborhood of $x = 0$. Again we can see that the system is not nilpotent. In fact,

$$\mathrm{ad}_{f_2}^{2n} f_1 = (-1)^n f_1 \quad n \geq 0 \ . \tag{13.8}$$

The results of the approximating procedure applied to this example are as followss (cf. [7]):

The cart is driven from $x_0 = (0,0,0,0)$ to $x_f = (0,-1,0,0)$. The computation of the controls uses the formulas explained in Examples 8.1 and 9.1. After some cancellations and regroupings the resulting controls are piecewise constant ($|u_i| \leq 1$) with 19 switches for each step, and they drive x_0 to x_f within an error of 0.07.

This system is also nilpotentizable and therefore a one step procedure can be used to reach the target exactly. The following feedback makes the system nilpotent.

$$\begin{aligned} u_1 &= \frac{1}{\cos(x_3)\cos(x_4)} w_1 \ , \\ u_2 &= \cos^3(x_4)\cos^2(x_3) w_2 - \frac{3}{l}\frac{\sin(x_4)\sin^2(x_3)}{\cos^2(x_4)} w_1 \ . \end{aligned} \tag{13.9}$$

In fact, if we make the change of coordinates

$$z_1 = x_1 \quad z_2 = \frac{\tan(x_3)}{\cos^3(x_4)} \quad z_3 = \tan(x_4) \quad z_4 = x_2 \ , \tag{13.10}$$

the system becomes,

$$\begin{aligned} \dot{z}_1 &= w_1 \\ \dot{z}_2 &= w_2 \ , \\ \dot{z}_3 &= \frac{1}{l} z_2 w_1 \ , \\ \dot{z}_4 &= z_3 w_1 \ , \end{aligned} \tag{13.11}$$

which is clearly nilpotent of order 3. Results of the simulation using a precompensator were given in [7]. Once again, the only errors are those due to numerical integration.

After the given change of coordinates the end points become $x_0 = (0,0,0,0)$ and $x_f = (0,0,0,-1)$. If f_1, f_2 are the vector fields for the nilpotent system, then the desired motion is exactly in the direction of $[f_1,[f_1,f_2]]$. The backward P. Hall coordinates are in fact $(0,-1,0,0,0)$. It is easily verified that the controls generate periodic trajectories for $\theta = x_4$ and $\psi = x_3$. While θ goes around once, ψ traverses two loops. This is a simple consequence of the identity $e^{[X,[X,Y]]} = e^X e^{[X,Y]} e^{-X} e^{-[X,Y]}$, which holds in $L_3(X,Y)$, together with the fact that the vectors $f_1, f_2, [f_1,f_2], [f_1,[f_1,f_2]]$ correspond to the coordinate axes x_1, x_3, x_4, x_2 respectively. In general, the trajectories prescribed by the forward P. Hall coordinates (which in this example are $(0,0,0,-1,0)$) are made up of moves in the directions of each bracket in the P. Hall basis. The move corresponding to a given bracket results in periodic motions of the variables corresponding to lower order brackets.

Finally, we study an example of a nilpotentizable system of higher order. Consider a front wheel drive cart as before but which in addition is pulling a trailer. The system has one more equation than before to account for the variable representing the angle the trailer makes with the x_1-axis.

The equations are,

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)\cos(x_4)u_1 \\ \dot{x}_2 &= \cos(x_3)\sin(x_4)u_1 \\ \dot{x}_3 &= u_2 \ , \\ \dot{x}_4 &= \frac{1}{l}\sin(x_3)u_1 \ , \\ \dot{x}_5 &= \frac{1}{d}\sin(x_4 - x_5)\cos(x_3)u_1 \ . \end{aligned} \tag{13.12}$$

The span of $\{f_1(x), f_2(x), [f_1,f_2](x), [f_1,[f_1,f_2]](x), [f_1,[f_1,[f_1,f_2]]](x)\}$ is $\mathbb{R}^5$ for x near zero. The system is nilpotentizable. In fact, it is feedback equivalent to a nilpotent system of order 4. To see this we proceed as in the example of the front wheel drive cart without a trailer and transform the equations into

$$\begin{aligned} \dot{z}_1 &= w_1 \ , \\ \dot{z}_2 &= w_2 \ , \\ \dot{z}_3 &= \frac{1}{l} z_2 w_1 \ , \\ \dot{z}_4 &= z_3 w_1 \ , \\ \dot{z}_5 &= \frac{1}{d}(z_3\cos(z_5) - \sin(z_5))w_1 \ . \end{aligned} \tag{13.13}$$

We now apply the feedback law

$$\begin{aligned} w_1 &= \bar{w}_1 \ , \\ w_2 &= \frac{\bar{w}_2 - \bar{g}(z_2, z_3, z_5)\bar{w}_1}{\frac{\partial \bar{h}}{\partial z_2}(z_2, z_3, z_5)} \ . \end{aligned} \tag{13.14}$$

where we defined $\bar{h}$ and $\bar{g}$ by

$$\begin{aligned} \bar{h}(z_2, z_3, z_5) &= \frac{1}{l} z_2 \cos z_5 - \frac{1}{d}(z_3 \sin z_5 + \cos z_5)(z_3 \cos z_5 - \sin z_5) \ , \\ \bar{g}(z_2, z_3, z_5) &= \frac{\partial \bar{h}}{\partial z_3}(z_2, z_3, z_5)\frac{1}{l} z_2 - \frac{\partial \bar{h}}{\partial z_5}(z_2, z_3, z_5)\frac{1}{d}(z_3 \sin z_5 + \cos z_5) \ . \end{aligned} \tag{13.15}$$

These functions $\bar{h}$ and $\bar{g}$ arise from trying to simplify the equations by making a change of coordinates $\bar{z}_3 = z_3 \cos z_5 - \sin z_5$. The differential equation for $\bar{z}_3$ is then

$$\dot{\bar{z}}_3 = \bar{h}(z_2, z_3, z_5)u_1 \ . \tag{13.16}$$

Moreover, since $\frac{\partial \bar{h}}{\partial z_2}(z_2, z_3, z_5) \neq 0$ (near $(0,0,0)$) we can also make the change $\bar{z}_2 = \bar{h}(z_2, z_3, z_5)$. Then the differential equation for $\bar{z}_2$ is

$$\dot{\bar{z}}_2 = \frac{\partial \bar{h}}{\partial z_2}(z_2, z_3, z_5)u_1 + \bar{g}(z_2, z_3, z_5)u_2 \ . \tag{13.17}$$

Therefore, with the feedback (13.14) and the change of coordinates

$$\begin{aligned} \bar{z}_1 &= z_1 \ , \\ \bar{z}_2 &= \bar{h}(z_2, z_3, z_5) \ , \\ \bar{z}_3 &= z_3 \cos z_5 - \sin z_5 \ , \\ \bar{z}_4 &= z_4 \ , \\ \bar{z}_5 &= z_5 \ . \end{aligned} \tag{13.18}$$

the system becomes

$$\begin{aligned} \dot{\bar{z}}_1 &= \bar{w}_1 \ , \\ \dot{\bar{z}}_2 &= \bar{w}_2 \ , \\ \dot{\bar{z}}_3 &= \bar{z}_2 \bar{w}_1 \ , \\ \dot{\bar{z}}_4 &= (\bar{z}_3 \sec \bar{z}_5 + \tan \bar{z}_5)\bar{w}_1 \ , \\ \dot{\bar{z}}_5 &= \frac{1}{d}\bar{z}_3 \bar{w}_1 \ . \end{aligned} \tag{13.19}$$

Notice that now $\dot{\bar{z}}_5$ does not depend on $\bar{z}_5$. The following feedback will now make the system nilpotent

$$\begin{aligned} \bar{w}_1 &= \tilde{w}_1 \ , \\ \bar{w}_2 &= \frac{\tilde{w}_2 - \tilde{g}(\bar{z}_2, \bar{z}_3, \bar{z}_5)\tilde{w}_1}{\frac{\partial \tilde{h}}{\partial \bar{z}_2}(\bar{z}_2, \bar{z}_3, \bar{z}_5)} \ . \end{aligned} \tag{13.20}$$

where the functions $\tilde{h}$ and $\tilde{g}$ are defined by

$$\tilde{h}(\bar{z}_2, \bar{z}_3, \bar{z}_5) = \frac{1}{d}\bar{z}_2 \sec^2 \bar{z}_5 + \frac{2}{d^2}\bar{z}_3 \sec^2 \bar{z}_5 \tan \bar{z}_5 \ , \tag{13.21}$$

$$\tilde{g}(\bar{z}_2, \bar{z}_3, \bar{z}_5) = \frac{\partial \tilde{h}}{\partial \bar{z}_3}(\bar{z}_2, \bar{z}_3, \bar{z}_5)\bar{z}_2 + \frac{\partial \tilde{h}}{\partial \bar{z}_5}(\bar{z}_2, \bar{z}_3, \bar{z}_5)\frac{1}{d}\bar{z}_3 \ , \tag{13.22}$$

These functions arise from making the change of coordinates $\tilde{z}_5 = \tan \bar{z}_5$ and propagating the changes as before. After these feedbacks and coordinate changes, and relabeling $\tilde{z}_1, \tilde{z}_2, \tilde{z}_3, \tilde{z}_4, \tilde{z}_5$ as x_1, x_2, x_3, x_5, x_4 and $\tilde{w}_1, \tilde{w}_2$ as u_1, u_2 the system equations become

$$\begin{aligned} \dot{x}_1 &= u_1 \ , \\ \dot{x}_2 &= u_2 \ , \\ \dot{x}_3 &= x_2 u_1 \ , \\ \dot{x}_4 &= x_3 u_1 \ , \\ \dot{x}_5 &= (x_3\sqrt{1 + x_4^2} + x_4)u_1 \ . \end{aligned} \tag{13.23}$$

§14. Conclusion .

The method presented above for solving the MPP rests on a sound theoretical basis, and our preliminary simulation results appear to show that it performs rather well in practice. Naturally, the procedure we have outlined can still be improved in a number of ways, and this requires continued research on several important issues.

A crucial problem is that of estimating the critical distance. The proof of Theorem 10.2 only provides a very rough estimate of $\Delta_c(\mathcal{R})$. In the simulations, it is clear that $\Delta_c(\mathcal{R})$ appears to be much larger. It would be desirable to have a way to estimate *a priori* a useful lower bound for $\Delta_c(\mathcal{R})$.

Another important question is that of the class of controls u being used. The method described here involves the concatenation of special pieces of a particular form, namely, controls that realize motion along the trajectories of the vector fields f_i. One can use bang-bang controls, but this is not necessary. For instance, the role of A in Example 9.1 can be played by a constant control $u_1 \equiv 1$, $u_2 \equiv 0$, during time 1, but we can also use instead a nonconstant control —e.g. $u_1(t) = \frac{1}{2}\sin t$, $0 \leq t \leq \pi$ — as long as its integral is equal to 1. (The same is true for B, of course.) In this way we can make sure that the controls we end up with are always continuous, in spite of the fact that they are obtained by concatenating simpler pieces. Alternatively, we can use more complicated pieces to build up our controls—e.g. controls that directly generate exponentials $e^{\alpha X + \beta Y}$. In this way, the total number of pieces can be cut down considerably.

A third important issue is that of fully exploiting the possibilities of feedback nilpotentization. This requires that one look for new classes of nilpotentizable systems, and also that one improve the existing nilpotentization results by making them as explicit as possible.

Finally, we point out that the methods presented here work because of the absence of drift. It certainly would be desirable to remove this restriction, but this will probably require a completely novel approach.

REFERENCES

1. Agrachev, A.A., and Gamkrelidze, R.V., "The exponential representation of flows and chronological calculus," *Math. Sbornik* **107** (149) (1978), pp. 467-532; Engl. Transl. in *Math. USSR Sbornik* **35** (1979), pp. 727-785.

2. Brockett, R.W., "Control theory and singular Riemannian geometry," in *New Directions in Applied Mathematics*, Springer-Verlag, New York (1981), pp. 11-27.

3. Fliess, M., "Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices non commutatives," *Invent. Math.*, **71** (1983), pp. 521-537.

4. Hauser, J., S. Sastry and P. Kokotovic, "Nonlinear control via approximate input-output linearization: the ball and beam example," *Proc. 28th IEEE CDC*, Tampa, Florida, Dec. 1989, pp. 1987-1993.

5. Hermes, H., A. Lundell and D. Sullivan, "Nilpotent bases for distributions and control systems," *J. Diff. Eqs.* **55**, No. 3 (1984), pp. 385-400.

6. Hermes, H., "Distributions and the Lie algebras their bases can generate," *Proc. Amer. Math. Soc.* **106**, No. 2 (1989), pp. 555-565.

7. Lafferriere, G., and H. J. Sussmann, "Motion planning for control systems without drift: a preliminary report," Rutgers Center for Systems and Control Technical Report 90-04, June 1990.

8. Murray, R.M., and S. S. Sastry, "Steering controllable systems," submitted to the 29th CDC.

9. Murray, R.M., and S. S. Sastry, "Grasping and manipulation using multifingered robot hands," Memorandum No. UCB/ERL M90/24, Electronics Research Laboratory, Univ. of California, Berkeley, March 1990.

10. Palais, R., "Global formulation of the Lie theory of transformation groups," *Mem. Amer. Math. Soc.* No. 22 (1957).

11. Sastry, S. and Zexiang Li, "Robot motion planning with nonholonomic constraints," *Proc. 28th IEEE CDC*, Tampa, Florida, Dec. 1989, pp. 211-216.

12. Sussmann, H.J., "Lie brackets and local controllability: a sufficient condition for scalar input systems," *S.I.A.M. J. Control and Optimization* **21**, No. 5 (1983), pp. 686-713.

13. Sussmann, H.J., "A product expansion for the Chen series," in *Theory and Applications of Nonlinear Control Systems*, C. Byrnes and A. Lindquist Eds., North-Holland, 1986, pp. 323-335.

14. Sussmann, H.J., "A general theorem on local controllability," *S.I.A.M. J. Control and Optimization*, **25**, No. 1 (1987), pp. 158-194.

15. Sussmann, H.J., and G. Lafferriere, "Motion planning for controllable systems without drift," in preparation.