# Trajectory Generation for Kinematic Legged Robots

Bill Goodwine and Joel Burdick
Division of Engineering and Applied Science
California Institute of Technology
Mail Code 104-44, Pasadena, CA 91125
Email: goodwine@@robby.caltech.edu, jwb@@robby.caltech.edu

**Abstract.** We present a general trajectory generation scheme for a class of "kinematic" legged robots. The method does not depend upon the number of legs, nor is it based on foot placement concepts. Instead, our method is based on an extension of a nonlinear trajectory generation algorithm for smooth systems to the legged case, where the relevant mechanics are not smooth. Our extension is based on the realization that legged robot configuration spaces are stratified. The algorithm is illustrated with a simple example.

## 1 Introduction

The trajectory generation problem for a legged robot is the problem of determining control inputs (*e.g.*, mechanism joint variable trajectories) which will steer the robot from a starting configuration to a desired final configuration. This paper presents a general trajectory generation scheme for a class of "kinematic" legged robots. The method is independent of the number of legs and is not based on foot placement concepts.

Our approach is motivated by the method of Lafferriere and Sussmann [1] for generating trajectories for a class of nonlinear "kinematic" systems whose equations of motion are smooth. However, since legged robots cyclically make and break contact with the ground, their equations of motion are non-smooth. Hence, the method of [1] can not be directly applied. We explicitly account for this non–smooth feature by a stratification of the robot's configuration space. Each strata correspond to configurations where different combinations of the feet are in ground contact. We extend the approach of [1] by decomposing a given problem into a set of trajectory generation problems on the individual strata, where a modification of their method can be used on each strata. We illustrate the application of this procedure with an example.

Our class of kinematic robots includes all quasistatic legged locomotors. One common component of trajectory generation for legged systems is the need to calculate foot placements, which can be computationally burdensome. In our approach, the focus is on control inputs, and the tricky issue of foot placement is avoided. Prior schemes have also been restricted to a particular number of legs (*e.g.* 4,6,8,$\cdots$). Our method works independently of the number of legs.

There is a vast literature on legged locomotion analysis, control, and motion planning. However, prior efforts have focused either on a particular morphology (*e.g.* biped [2], quadruped [3], or hexaped [4]) or a particular locomotion assumption (*e.g.* quasi-static [4] or hopping [5]). Less effort has been devoted to uncovering principles that span all morphologies and assumptions. Future robotic engineers will want a locomotion "mechanics" and "control" theory that is general (*i.e.* not constrained to a particular morphology), implementable in an automated way (*i.e.*, general software "tool-kits" can developed), and that has verifiable and provable properties. Some recent works have attempted to uncover some of the fundamental structure underlying locomotion mechanics. Kelly and Murray [6] showed that a number of "kinematic" locomotive systems can be modeled using connections on principal fiber bundles. They also provide results on controllability, as well as an interpretation of movement in terms of geometric phases. Ostrowski [7], [8] developed analogous results for a class of "dynamic" nonholonomic locomotion systems. The results presented in this paper are a small contribution to the control aspect of our envisioned engineering basis. Our work makes a novel connection with recent advances in nonlinear geometric control theory. We believe that this connection is a useful and necessary step towards establishing a solid basis for locomotion engineering. We note that Tsakiris and Krishnaprasad [9] have used methods from nonlinear control theory to develop motion planning schemes for "G"-snakes, a class of kinematic undulatory mechanisms.

## 2 Background

To set the context for our approach, this section outlines the method in [1] for the generation of trajectories for smooth, kinematic nonholonomic systems which are in the form of a non-linear affine driftless system evolving on a configuration manifold, $M$

$$\dot{x} = g_1(x)u_1 + \cdots + g_m(x)u_m \qquad x \in M. \tag{1}$$

Nonholonomic systems typically do not have enough controls to directly drive each state variable along a given trajectory. The trajectory generation problem for systems with such a deficit is managed by using an "extended system," where "fictitious controls," corresponding to higher order Lie bracket motions, are added.

Recall that the Lie bracket between two control vector fields, $g_1(x)$ and $g_2(x)$, is computed as

$$[g_1(x), g_2(x)] = \frac{\partial g_2(x)}{\partial x} g_1(x) - \frac{\partial g_1(x)}{\partial x} g_2(x)$$

and can be interpreted as the leading order term that results from the sequence of flows

$$\phi_\epsilon^{g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{-g_2}(x) = \epsilon^2 [g_1, g_2](x) + \mathcal{O}(\epsilon^3),$$

where $\phi_\epsilon^{g_1}(x_0)$ represents the solution of the differential equation $\dot{x} = g_1(x)$ at time $\epsilon$ starting from $x_0$.

The trajectory generation problem is relatively easy to solve for the extended system. The real controls are then computed from the fictitious controls associated with the extended system. We first must review two related concepts before summarizing the approach. More details on both concepts can be found in Serre [10]. We assume that the reader is familiar with the basic concepts of nonlinear geometric control theory, as in [11].

**The Philip Hall basis.** Because of Jacobi's identity and the fact that a Lie bracket is skew symmetric, it is not easy to select a basis for the Lie algebra generated by the set of vector fields $\{g_i(x)\}$ in Equation 1. A *Philip Hall basis* is a particular way to select a basis. Given a set of vector fields $\{g_1, \ldots, g_m\}$, define the *length* of a *Lie product* as

$$\begin{aligned} l(g_i) &= 1 \qquad i = 1, \ldots, m \\ l([A, B]) &= l(A) + l(B), \end{aligned}$$

where $A$ and $B$ may be Lie products. A *Philip Hall basis* is an ordered set of Lie products $H = \{B_i\}$ satisfying:

1. $g_i \in H, i = 1, \ldots, m$
2. If $l(B_i) < l(B_j)$, then $B_i < B_j$
3. $[B_i, B_j] \in H$ if and only if
   (a) $B_i, B_j \in H$ and $B_i < B_j$ and
   (b) either $B_j = g_k$ for some $k$ or $B_j = [B_l, B_r]$ with $B_l, B_r \in H$ and $B_l \leq B_i$.

**The Campbell–Hausdorff Formula.** The flow along $g_i$ is referred to as the *formal exponential* of $g_i$ and is denoted

$$\phi_t^{g_i}(x) := e^{tg_i}(x) = (I + tg_i + \frac{t^2}{2} g_i^2 + \cdots). \qquad (2)$$

We note that terms of the form $g_i^2$ must be carefully justified. This is done by associating a Lie algebra of *indeterminates* to the Lie algebra of vector fields associated with the control problem. A complete exposition on the use of indeterminates in this manner is omitted here due to space limitations and we refer the reader to references [1], [12], [13] for a complete explanation.

We note that in order to use Equation 2, composition must be from left to right, as opposed to right to left for flows, *e.g.*,

$$\phi_{t_2}^{g_2} \circ \phi_{t_1}^{g_1} = e^{g_1 t_1} e^{g_2 t_2},$$

which means "flow along $g_1$ for time $t_1$ and then flow along $g_2$ for time $t_2$." The Campbell–Baker–Hausdorff formula is given in the following theorem.

THEOREM 2.1 *Given two smooth vector fields $g_1, g_2$ the composition of their exponentials is given by*

$$e^{g_1} e^{g_2} = e^{g_1 + g_2 + \frac{1}{2}[g_1, g_2] + \frac{1}{12}([g_1,[g_1,g_2]] - [g_2,[g_1,g_2]]) \cdots}$$

*where the remaining terms may be found by equating terms in the (non-commutative) formal power series on the right– and left–hand sides.*

**Trajectory Generation for Smooth Systems.** In this section, we limit our attention to a smooth, kinematic system described by a single set of equations of motion having the form of Equation 1. Such a system is said to be *nilpotent of order k* if all the Lie brackets between control vector fields of order greater than $k$ are 0. The method presented in this section works exactly for nilpotent systems, and approximately for systems which are not nilpotent. For non-nilpotent systems, arbitrary precision can be obtained by iterating the algorithm.

Associate with the system in Equation 1 the *extended system*:

$$\dot{x} = g_1 v^1 + \cdots g_m v^m + g_{m+1} v^{m+1} + \cdots + g_s v^s \qquad (3)$$

where the $g_{m+1}, \ldots, g_s$ are higher order Lie brackets of the $g_i$, chosen so that $\dim(\text{span}\{g_1, \ldots, g_s\}) = \dim(M)$. The $v^i$'s are called *fictitious inputs* since they may not correspond with the actual system inputs. The higher order Lie brackets must belong to the Philip Hall basis for the Lie algebra. The control inputs $v^i$ which steer the extended system can be found as follows. To go from a point $p$ to a point $q$, define a curve, $\gamma(t)$ connecting $p$ and $q$ (a straight line would work, but is not necessary). After determining $\gamma$, simply solve

$$\dot{\gamma}(t) = g_1(\gamma(t)) v^1 + \cdots + g_s(\gamma(t)) v^s \qquad (4)$$

for the fictitious controls $v_i$. This will involve inverting a square matrix or determining a pseudo–inverse, depending on whether or not there are more $g_i$'s than the dimension of the configuration space.

The actual control inputs can be found as follows. Determine the Philip Hall basis for the Lie algebra generated by $g_1, \ldots, g_m$, and denote it by $B_1, B_2, \ldots, B_s$. It is possible to represent all flows of Equation 1 in the form

$$S_t(x) = e^{h_s(t) B_s} e^{h_{s-1}(t) B_{s-1}} \cdots e^{h_2(t) B_2} e^{h_1(t) B_1}(x)$$

for some functions $h_1, h_2, \ldots, h_s$, called the (backward) Philip Hall coordinates. Furthermore, $S_t(x)$ satisfies the differential equation

$$\dot{S}(t) = S(t)(B_1 v_1 + \cdots + B_s v_s); \quad S(0) = 1, \tag{5}$$

where $S_t(x)$ has been replaced by $S(t)$. If we define the *adjoint mapping*

$$\mathrm{Ad}_{e^{-h_i B_i}} B_j = e^{-h_i B_i} B_j e^{h_i B_i},$$

then it can be shown that

$$\mathrm{Ad}_{e^{-h_i B_i} \cdots e^{-h_{j-1} B_{j-1}}} B_j \dot{h}_j = \left( \sum_{k=1}^{s} p_{j,k}(h) B_k \right) \dot{h}_j, \tag{6}$$

for some polynomials $p_{j,k}(h)$. (For a more detailed explanation, see [14]). Equating coefficients yields the differential equations

$$\dot{h} = A(h) v \qquad h(0) = 0. \tag{7}$$

These equations specify the evolution of the Philip Hall coordinates in response to the fictitious inputs, which were found via Equation 4. Next we determine the actual inputs using the Philip Hall coordinates.

It is easier to determine the real inputs using the forward rather than backward Philip Hall coordinates. The transformation from the backward to forward coordinates is a "simple algebraic transformation" (see [1]), and this transformation results in an equation of the form

$$S = e^{h_1 B_1} e^{h_2 B_2} \cdots e^{h_{s-1} B_{s-1}} e^{h_s B_s}.$$

In this paper, we avoid the need for this transformation between forward and backward Philip Hall coordinates by limiting our attention to systems which are nilpotent of order two. The process to determine the actual inputs is best illustrated by a simple example.

Example 2.2 (from [1]). Find $u$ for $S(T) = e^{\alpha X} e^{\beta Y} e^{\gamma [X,Y]} e^{\delta [X,[X,Y]]} e^{\epsilon [Y,[YX,Y]]}$.

Recall that composition is from left to right. So, first, determine the control input for the first two terms, and call them $\alpha A$ and $\beta B$. We will use # for concatenation, so $\alpha A \# \beta B$ means control $\alpha A$ followed by control $\beta B$. Clearly, $\alpha A \# \beta B$ gives rise to system evolution $e^{\alpha X} e^{\beta Y}$.

Now, we want to steer in the Lie bracket direction $\gamma[X, Y]$. To leading order, the sequence $\sqrt{\gamma} A \# \sqrt{\gamma} B \# (-\sqrt{\gamma} A) \# (-\sqrt{\gamma} B)$ gives rise to $e^{\gamma [X,Y]}$. However, using the Campbell–Baker–Hausdorff formula, and assuming a nilpotency of degree three, it actually gives rise to

$$e^{\gamma [X,Y]} e^{\frac{1}{2} \gamma^{\frac{3}{2}} [X,[X,Y]]} e^{-\frac{1}{2} \gamma^{\frac{3}{2}} [Y,[X,Y]]}.$$

We now must find a control that gives rise to $e^{(\delta - \frac{1}{2} \gamma^{\frac{3}{2}})[X,[X,Y]]} e^{(\epsilon + \frac{1}{2} \gamma^{\frac{3}{2}})[Y,[X,Y]]}$, to cancel out the third order error from above and give the correct third order exponentials. The 20 sequence move is presented in [1] and is easy to determine using the Campbell–Baker–Hausdorff formula.

The basic idea is that we construct the control inputs that give us the appropriate first order flow, and then take care of the higher order error when constructing the control input sequence to execute the Lie bracket directions. We note furthermore, that if the system is nilpotent of order 2, then there is no need to transform to the forward coordinates because the higher order error will not be present.

This method generates the actual control inputs necessary to follow the desired trajectory. If the system is nilpotent, [1] proves that this method exactly steers the system to the desired final state. If the system is not nilpotent, they show that it steers it to a point that is, at worst, only half the distance to the desired configuration. The algorithm can thus be iterated to generate arbitrary precision. Finally, we note that the results in [1] also include the notion of a "critical" step length. While Lafferiere and Sussmann prove the existence of a critical step and give a rough bound for it, they acknowledge that a better means to estimate the critical length is desirable. An appropriate step length can be determined by simulation or experiment, and the simulation results in [1] show that the actual critical length is larger than the estimated bound.

## 3 Stratified Configuration Spaces

The motion planning approach of [1] can not be directly applied to trajectory generation for legged robots because their equations of motion are not smooth. However, there is sufficient structure inherent in legged robot configuration spaces to develop a method based upon the results reviewed in Section 2.

The configuration space (*c-space*) for a kinematic, legged robot is *stratified.* (See [15] for details on stratifications). A *regularly stratified* set $\mathcal{X}$ is a set $\mathcal{X} \subset \mathbb{R}^m$ decomposed into a finite union of disjoint smooth manifolds, called *strata*, satisfying the Whitney condition. The dimension of the strata varies between zero, which are isolated point manifolds, and $m$, which are open subsets of $\mathbb{R}^m$. The Whitney condition requires that the tangents of two neighboring strata "meet nicely," and for our purposes it suffices to say that this condition is generically satisfied.

Legged robot c-spaces have even more structure that arises from their multi-legged nature. Let $M$ denote the legged robot's entire configuration manifold (it will often be convenient to denote this space as $S_0$). Let $S_i \subset M$ denote the codimension one submanifold of $M$ that corresponds to all configurations where only

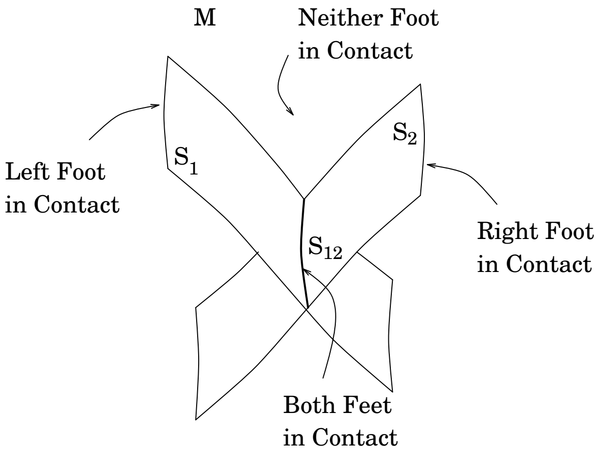

**Figure 1.** Biped Configuration Space

the $i^{th}$ foot contacts the terrain. Generically, the intersection of $S_i$ and $S_j$, denoted $S_{ij} = S_i \cap S_j$, will be a submanifold with codimension two. The set $S_{ij}$ physically corresponds to states where the $i^{th}$ and $j^{th}$ feet are on the ground. Higher codimension submanifolds can be similarly defined in a recursive fashion: $S_{ijk} = S_i \cap S_j \cap S_k = S_i \cap S_{jk}$, etc. Figure 1 pictorially illustrates such a structure for a biped robot. Technically speaking, strata $\mathcal{X}_i$ consists of the submanifold $S_i$ with all lower dimensional strata (that arise from intersections of $S_i$ with other submanifolds) removed. However, by abuse of notation, we will often refer to the submanifolds $S_i$, as well as their recursive intersections $S_{ij}, S_{ijk}$, etc, as strata. We will term the highest codimension strata containing the point $x$ as the *bottom strata*, and any other submanifolds containing $x$ as *higher strata.*

Whenever an additional foot contacts the ground, the robot is subjected to additional constraints. For "point–like" feet, this may be a holonomic constraint; whereas, other foot structures may be better characterized by a "rolling without slipping" constraint. Regardless of the particular form of the additional constraint, it will change the system's equations of motion in a non–smooth manner. This non–smooth transition will occur whenever the system transitions between any two submanifolds or strata.

We assume that robot's equations of motion are smooth when restricted to a strata, and undergo a non–smooth change only when transitioning between strata. Let $I$ denote an index set: $I = \{i_1, \cdots, i_k\}$. Let $S_I = S_{i_1,\cdots,i_k}$ denote the submanifold whose describing indices are contained in $I$. The robot's equations of motion at a point $x \in S_I$ will be denoted as

$$\dot{x} = g_{S_I,1}(x)u^{S_I,1} + \cdots + g_{S_I,p_I}(x)u^{S_I,p_I}, \qquad (8)$$

where $g_{S_I,j}$ is the $j^{th}$ control vector field and $p_I$ is the number of control inputs when the system is constrained to $S_I$. The number of control inputs may differ among strata because the physical constraints associated with a particular submanifold may additionally constrain the control inputs or the robot's kinematic movement. Thus, the control input $u^{M,1}$ does not necessarily correspond to the control input $u^{S_I,1}$.

For the biped robot whose configuration space is depicted in Figure 1, there will be four strata and three or four sets of motion equations. One set will describe (on submanifold $S_{12}$) the evolution of the system when both feet are in contact with the ground. Two sets of equations will describe (on submanifolds $S_1$ and $S_2$) the system when individual feet make ground contact. A fourth set of equations on $M$ will describe the state where both feet simultaneously lift off the ground. This fourth set of equations will typically be disallowed for kinematic systems.

For a given strata, $S_I$, the *distribution* defined by the span of the control vector fields active on $S_I$ is:

$$\Delta_{S_I} = \mathrm{span}\{g_{S_I,1}, \ldots, g_{S_I,p_I}\}.$$

The *involutive closure* of $\Delta_{S_I}$, denoted by $\overline{\Delta}_{S_I}$ is the closure of $\Delta_{S_I}$ under Lie bracketing, and it will be called a *controllability Lie algebra* (see [11] for more details). A basic assumption in this paper is that the robot is controllable, which practically implies that it is possible to generate control inputs that will move the system in any direction (though these control inputs may involve complicated switching between strata). In [16] the authors have developed a procedure to test the controllability of a stratified kinematic system, and this assumption of controllability, in certain simple cases, can be stated as

$$\overline{\Delta}_M + \sum_i \overline{\Delta}_{S_i} + \sum_{i,j} \overline{\Delta}_{S_{ij}} + \cdots = T_x M. \qquad (9)$$

## 4 Legged Trajectory Generation

In this section, we extend the procedure outlined in Section 2 to kinematic legged systems with a stratified c-space. Assume that the robot starts at configuration $p$ and seeks to reach a final configuration $q$. Assume that the system is nilpotent to degree $k_{S_I}$ (for strata $S_I$), or that we are making a nilpotent approximation to degree $k_{S_I}$. Each strata may have a different degree of nilpotency or nilpotent approximation. Finally, assume for simplicity that points $p$ and $q$ lie in the same bottom strata. In the biped example, this assumption states that if the robot starts with both feet in contact with the ground, it must end with both feet in contact. Eliminating this requirement would be a straightforward extension of this algorithm.

Assume that $p, q \in S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}$, so that the bottom strata is $S_I$. Assume also that we are restricted to the collection of submanifolds $\{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\}$, *i.e.*, lifting all the legs off the ground at once is not allowed. The stratified trajectory generation algorithm is comprised of the following steps:

1. Construct the extended systems (Section 2) on $S_I$ and higher strata which contain $p$ and $q$.
2. Find a curve, $\gamma(t)$, connecting $p$ and $q$. In general, $\gamma(t)$ will need to switch among the strata which contain $p$ and $q$. To do this, we must consider the *stratified extended system*, discussed later.
3. Solve the stratified extended system for the fictitious inputs.
4. For each segment in each strata: compute the backward Philip Hall coordinates; if necessary, transform them into forward Philip Hall coordinates; compute (strata–by–strata) the control inputs that steer the system along $\gamma(t)$.

Each step is a relatively straightforward application of the results discussed in Section 2, except for Step 2. The notion of a "stratified extended system," is the key to implementing Step 2, and is the key concept that extends the results of Section 2 to legged robots.

As mentioned above, the path $\gamma(t)$ that connects $p$ to $q$ must generally switch among multiple strata. This switching behavior, which can not be accounted for in the method of Section 2, can be incorporated as follows. On each strata, only one set of controls is in effect, and the equations of motion in the bottom strata will be different than in the higher strata. However, since the bottom strata is defined by the intersection of higher strata, the equations of motion in the higher strata are valid at points arbitrarily close to the bottom strata. Hence, the general approach to determine all of the control in the various neighboring strata is to consider the equations of motion in each strata simultaneously, assuming that each is valid at points in the bottom strata. The *stratified extended system* has the form

$$\dot{x} = \tilde{g}_1 v^1 + \tilde{g}_2 v^2 + \cdots + \tilde{g}_p v^k \qquad (10)$$

where the vector fields $\tilde{g}_i$ are all the vector fields from all the extended systems associated with each strata computed in Step 1.

Recall from Section 2 that to find the fictitious control inputs we solved the extended system containing the extended control vector fields.

$$\dot{\gamma}(t) = g_1(\gamma(t))v^1 + \cdots + g_m(\gamma(t))v^m. \qquad (11)$$

The solution to this equation will yield control inputs *which must be executed simultaneously.* Since the robot switches among strata to execute control motions, it is impossible to simultaneously execute the control inputs that are computed from Equation 11 if these inputs are associated with different strata. The control inputs associated with different strata must be executed sequentially. This fact will lead to an error, as can be seen in the following example. Assume that Equation 11 gives two inputs, $v^1(t)$ and $v^2(t)$. The flow resulting from the simultaneous execution of $v^1(t)$ and $v^2(t)$ is $e^{g_1 v^1 + g_2 v^2}$. However, if we execute $v^1$ and then $v^2$ sequentially, the Campbell–Baker–Hausdorff formula gives

$$e^{g_1 v^1} e^{g_2 v^2} = e^{g_1 v^1 + g_2 v^2 + [g_1 v^1, g_2 v^2] + \cdots}$$

which is clearly not the same flow.

Frequent switching between strata will minimize this error since it more closely approximates the action of simultaneous control execution, and thereby reduces the higher order difference between the simultaneous and sequential flows. This rapid switching, which physical corresponds to many small footsteps, may not be desirable. In certain "decoupled" cases rapid switching may not be necessary. Control vector fields associated with different strata are *strata decoupled* when the Lie bracket between any two vector fields belonging to different strata is zero. In that case the Campbell–Baker–Hausdorff formula shows that the simultaneous flow and sequential flow are equal. Extending the results of this paper to construct a switching algorithm which requires neither rapid switching nor strata decoupling will be the subject of a future publication. We will hereafter assume that the equations of motion are strata decoupled. As the example in Section 5 illustrates, this assumption may be commonly satisfied.

## 5 Example

We illustrate the application of our approach by generating control inputs which will steer a hexapod robot model (Figure 2, which is adapted from [6]). Assume that the robot walks with a tripod gait, alternating movements of legs 1–4–5 with movements of legs 2–3–6. The c-space of this hexapod is abstractly similar to that of the biped illustrated in Figure 1. This robot has four control inputs. The first two inputs, $u^1$ and $u^2$ control the forward and backward angular leg displacements. The first control input controls the movement of legs 1–4–5, and the second control legs 2–3–6. The other two inputs, $u^3$ and $u^4$ control the height of legs 1–4–5 and 2–3–6 respectively.

The equations motion in the bottom strata, $S_{12}$ (where all the feet maintain ground contact), are:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ l & -l \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix} \qquad (12)$$

where $(x, y, \theta)$ represents the planar position of the

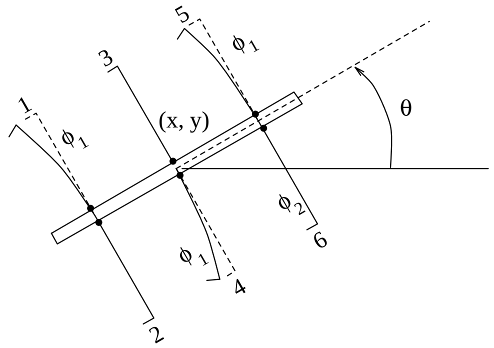

**Figure 2.** Six Legged Robot

robot's center. $\phi_1$ is the angle of legs 1–4–5 and $\phi_2$ is the angle of legs 2–3–6. The variables $u^3$ and $u^4$ are constrained to be 0 (so that the legs maintain ground contact). If $g_{S_{12},1}$ and $g_{S_{12},2}$ represent the first and second columns in Equation 12, then

$$\begin{aligned} g_{S_{12},3} &= [g_{S_{12},1}, g_{S_{12},2}] \\ &= (-(2l\sin\theta)\ (2l\cos\theta)\ 0\ 0\ 0)^T\ . \end{aligned} \tag{13}$$

If legs 1–4–5 are in contact with the ground, but legs 2–3–6 are not in contact, the equations of motion are

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ l & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ u^4 \end{pmatrix} \tag{14}$$

where $h_i$ is the height of the corresponding set of legs and $u^3$ is constrained to be 0. Call columns one, two and three in Equation 14 $g_{S_1,1}, g_{S_1,2}$ and $g_{S_1,3}$, respectively. This higher strata will be called $S_1$. If legs 2–3–6 are in ground contact and legs 1–4–5 are not, the equations of motion are

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \end{pmatrix} = \begin{pmatrix} 0 & \cos\theta & 0 \\ 0 & \sin\theta & 0 \\ 0 & -l & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ u^3 \end{pmatrix} \tag{15}$$

where $u^4$ is constrained to be 0. The columns in Equation 15 will be denoted $g_{S_2,1}, g_{S_2,2}$ and $g_{S_2,3}$, respectively, and this higher strata is $S_2$.

It is simple to verify that these equations of motion satisfy the controllability requirement expressed in Equation 9.

The extended system on each strata is constructed in the first step of the algorithm. On strata $S_{12}$, simple calculations verify that

$$\overline{\Delta}_{S_{12}} = \operatorname{span}\{g_{S_{12},1}, g_{S_{12},2}, g_{S_{12},3}\}$$

is involutive. Thus, the extended system on $S_{12}$ is

$$\dot{x} = g_{S_{12},1}v^{S_{12},1} + g_{S_{12},2}v^{S_{12},2} + g_{S_{12},3}v^{S_{12},3}. \tag{16}$$

This Lie algebra is *not* nilpotent, and thus the extended system of Equation 16 is only a nilpotent approximation. The inclusion of higher order terms (*i.e.*, third, fourth or even higher order Lie brackets) would result in a better approximation. For the simulation results presented below, only Equation (16) was used. Note also that $g_{S_{12},1}, g_{S_{12},2}$ and $g_{S_{12},3}$ constitute a Philip Hall basis up to degree two.

On strata $S_1$, it is easy to verify that $[g_{S_1,i}, g_{S_1,j}] = 0$ $\forall i, j$. Thus,

$$\overline{\Delta}_{S_1} = \operatorname{span}\{g_{S_1,1}, g_{S_1,2}, g_{S_1,3}\}$$

is involutive and nilpotent of order 1. The extended system on strata $S_1$ is:

$$\dot{x} = g_{S_1,1}v^{S_1,1} + g_{S_1,2}v^{S_1,2} + g_{S_1,3}v^{S_1,3}.$$

Similarly, on strata $S_2$ it is easy to verify that $[g_{S_2,i}, g_{S_2,j}] = 0$ $\forall i, j$. Thus,

$$\overline{\Delta}_{S_2} = \operatorname{span}\{g_{S_2,1}, g_{S_2,2}, g_{S_2,3}\}$$

is involutive and nilpotent of order 1. Thus, on strata $S_2$, the extended system is

$$\dot{x} = g_{S_2,1}v^{S_2,1} + g_{S_2,2}v^{S_2,2} + g_{S_2,3}v^{S_2,3}.$$

Thus, the stratified extended system is

$$\begin{aligned} \dot{x} &= g_{S_{12},1}v^{S_{12},1} + g_{S_{12},2}v^{S_{12},2} + g_{S_{12},3}v^{S_{12},3} \\ &+ g_{S_1,1}v^{S_1,1} + g_{S_1,2}v^{S_1,2} + g_{S_1,3}v^{S_1,3} \\ &+ g_{S_2,1}v^{S_2,1} + g_{S_2,2}v^{S_2,2} + g_{S_2,3}v^{S_2,3}. \end{aligned}$$

However, since $g_{S_{12},1} = g_{S_1,1}$ and $g_{S_{12},2} = g_{S_2,2}$, the stratified extended system can be rearranged to

$$\begin{aligned} \dot{x} &= g_{S_{12},1}v^{S_{12},1} + g_{S_{12},2}v^{S_{12},2} + g_{S_{12},3}v^{S_{12},3} \\ &+ g_{S_1,2}v^{S_1,2} + g_{S_1,3}v^{S_1,3} \\ &+ g_{S_2,1}v^{S_2,1} + g_{S_2,3}v^{S_2,3}. \end{aligned}$$

One can verify that this system is strata decoupled.

Let the starting and ending configurations be:

$$\begin{aligned} p &= (x, y, \theta, \phi_1, \phi_2, h_1, h_2) = (0, 0, 0, 0, 0, 0, 0) \\ q &= (x, y, \theta, \phi_1, \phi_2, h_1, h_2) = (1, 1, 0, 0, 0, 0, 0) \end{aligned}$$

A path that connects these points is $\gamma(t) = (t, t, 0, 0, 0, 0, 0)$. Equating $\dot{\gamma}(t)$ with with the stratified extended system and solving for the fictitious controls yields:

$$\begin{pmatrix} v^{S_{12},1} \\ v^{S_{12},2} \\ v^{S_{12},3} \\ v^{S_1,2} \\ v^{S_1,3} \\ v^{S_2,1} \\ v^{S_2,3} \end{pmatrix} = \frac{1}{2l} \begin{pmatrix} l(\cos\theta + \sin\theta) \\ l(\cos\theta + \sin\theta) \\ \cos\theta - \sin\theta \\ -l(\cos\theta + \sin\theta) \\ 0 \\ -l(\cos\theta + \sin\theta) \\ 0 \end{pmatrix}.$$

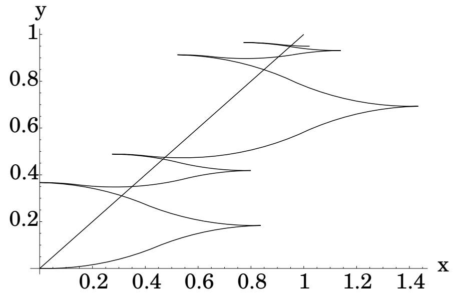

**Figure 3.** Straight Trajectory

For a system which is nilpotent of order 2, we have from Equation 6 for the extended system on $S_{12}$

$$\begin{aligned}\dot{h}_1 &= v^{S_{12},1}, \quad \dot{h}_2 = v^{S_{12},2}, \\ \dot{h}_3 &= v^{S_{12},3} + h_1, v^{S_{12},2}\end{aligned}$$

which yields.

$$h_1(1) = \frac{1}{2} \quad h_2(1) = \frac{1}{2} \quad h_3(1) = \frac{3}{4}. \qquad (17)$$

Since the nilpotent approximation is of order two, there is no need to transform to forward Philip Hall coordinates. The control sequence is

$$\sqrt{\frac{3}{4}}(v^{S_{12},1}\# v^{S_{12},2}\# - v^{S_{12},1}\# - v^{S_{12},2})$$

to get $e^{h_3 B_3}$, and $\frac{1}{2}v^{S_{12},2}\# v^{S_{12},1}$ to get $e^{h_2 B_2}e^{h_1 B_1}$. Hence, the complete sequence is

$$\sqrt{\frac{3}{4}}(v^{S_{12},1}\# v^{S_{12},2}\# - v^{S_{12},1}\# - v^{S_{12},2})\#\frac{1}{2}v^{S_{12},2}\# v^{S_{12},1}.$$

On the higher strata $S_1$, the extended system is:

$$\dot{h}_1 = v^{S_1,2} \quad \dot{h}_2 = v^{S_1,3} \qquad (18)$$

which gives

$$h_1(1) = -\frac{1}{2} \quad h_2(1) = 0. \qquad (19)$$

This dictates a control sequence $-\frac{1}{2}v^{S_1,2}$. Similarly, the extended system on $S_2$ gives the control sequence $-\frac{1}{2}v^{S_2,3}$. Thus, the total control sequence is

$$\begin{aligned}&\sqrt{\tfrac{3}{4}}(v^{S_{12},1}\# v^{S_{12},2}\# - v^{S_{12},1}\# - v^{S_{12},2}) \\ &\#\tfrac{1}{2}v^{S_{12},2}\# v^{S_{12},1}\#(-\tfrac{1}{2}v^{S_1,2})\# - (\tfrac{1}{2}v^{S_2,3}).\end{aligned}$$

Figure 3 shows the path of the robot's center as it follows a straight line trajectory, which is broken into four equal segments. Due to the nilpotent approximation, there is some small final error. Better accuracy can be obtained by use of a higher order nilpotent approximation or a second iteration of the algorithm from the robot's ending position.

The approach is general enough that arbitrary smooth trajectories are possible. Figure 4 shows the the hexapod tracking an ellipse while maintaining a constant angular orientation. Figure 5 shows the results when a smaller step size is used. In this example, part of the trajectory tracking error is due to the nilpotent approximation, but another contribution to the trajectory error is the simplicity of the model. Essentially, some directions are more "difficult" for the system to execute than others. In these simulations, because of the simplicity of the model, which eliminates "crab–like" gates, when the robot has to move sideways, its tracking error is greater because this direction corresponds to a Lie bracket direction.

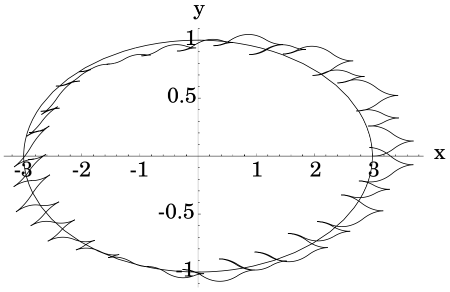

**Figure 4.** Elliptical Trajectory

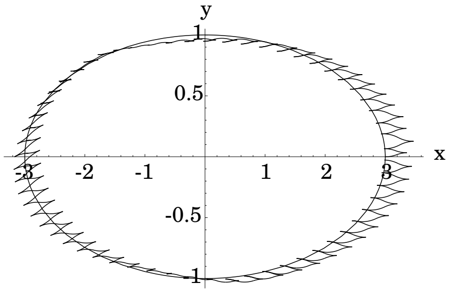

**Figure 5.** Elliptical Trajectory

Figure 6 shows the hexapod following the same ellipse while also rotating at a constant rate. Figure 7 plots the robot's angular orientation as the simulation progresses.

## 6 Conclusions

Our method provides a general means to solve the trajectory generation problem for certain types of legged robotic systems and the simulations indicate that the approach is rather simple to apply. The method is independent of the number of legs and is not based on foot placement principles. For a given legged robot mechanism, the deployment of a specificly tuned leg-placement-based algorithm may lead to motions which use fewer steps or results in less tracking error. However, for the purposes of initial design and evaluation of a legged mechanism, our approach affords the robotic design engineer an automated way to implement a realistic trajectory generation scheme

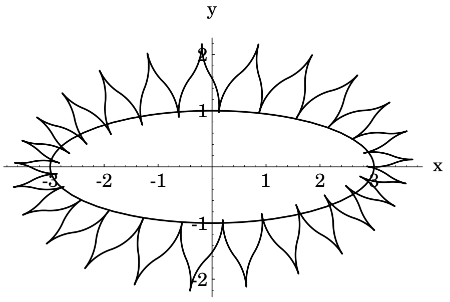

**Figure 6.** Elliptical Path with Rotation

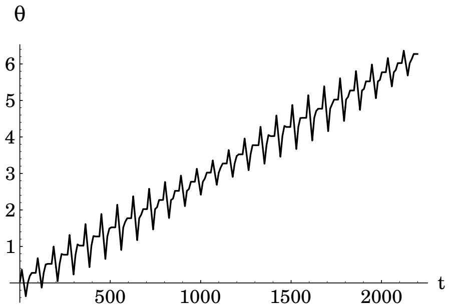

**Figure 7.** Hexapod Orientation

for a quasi-static robot of nearly arbitrary morphology. More importantly, we believe that are approach provides an evolutionary path for future research and generalizations. Clearly, this general framework also encompasses other types of systems whose configuration space is similarly stratified. An obvious example would be a robotic grasping problem, where we wish to reorient an object grasped by a robot hand by used of repeated finger repositioning.

There are several avenues of potential further work. In this paper we addressed the simple case where the equations of motion are strata decoupled. While experience dictates that many robots are strata decoupled, the non-decoupled case merits further attention. Since many of the most interesting types of robotic systems (such as bipeds) are not kinematic, an algorithm for solving the trajectory generation problem for such systems is necessary. However, since the state of the art for solving the trajectory generation problem for smooth systems with drift is still in its infancy (see, for example, the special results in References [17], and the references cited therein), it may be difficult to make headway along these lines until more complete results for the smooth case become known.

**Acknowledgments:** This work was partially supported by a grant from the Office of Naval Research.

## References

[1] G. Lafferriere and Hector J. Sussmann. A differential geometric approach to motion planning. In X. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 235–270. Kluwer, 1993.

[2] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain. In *IEEE International Conference Robotics and Automation*, pages 1405–1411, Sacramento, CA, 1991.

[3] J.K. Lee and S.M. Song. Path planning and gait of walking machine in an obstacle-strewn environment. *J. Robotics Systems*, 8:801–827, 1991.

[4] S.M. Song and K.J. Waldron. *Machines that walk: the Adaptive Suspension Vehicle.* MIT Press, 1989.

[5] M.H. Raibert. *Legged Robots that Balance.* MIT Press, 1986.

[6] Scott D. Kelly and Richard M. Murray. Geometric phases and robotic locomotion. *J. Robotic Systems*, 12(6):417–431, 1995.

[7] J. Ostrowski, J. Burdick, R. Murray, and A. Lewis. The mechanics of undulatory locomotion: the mixed dynamic and kinematic case. In *IEEE International Conference on Robotics and Automation*, pages 1945–1951, Nagoya, Japan, 1995.

[8] J.P. Ostrowski. *The Mechanics and Control of Undulatory Robotic Locomotion.* PhD thesis, California Institute of Technology, Pasadena, CA, 1995.

[9] P. S. Krishnaprasad and D. P. Tsakiris. *G*-snakes: Nonholonomic kinematic chains on Lie groups. In *IEEE International Conference on Decision and Control*, pages 2955–2960, Lake Buena Vista, FL, 1994.

[10] Jean-Pierre Serre. *Lie Algebras and Lie Groups.* Springer–Verlag, 1992.

[11] Alberto Isidori. *Nonlinear Control Systems.* Springer–Verlag, second edition, 1989.

[12] Hector J. Sussmann. Lie brackets and local controllability: A sufficient condition for scalar–input systems. *Siam J. Control and Optimization*, 21(5):686–713, 1983.

[13] Hector J. Sussmann. A general theorem on local controllability. *Siam J. Control and Optimization*, 25(1):158–194, 1987.

[14] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., 1994.

[15] Goresky and Macpherson. *Stratified Morse Theory.* Springer–Verlag, New York, 1980.

[16] Bill Goodwine and Joel Burdick. Controllability of kinematic control systems on stratified configuration spaces. preprint, 1997.

[17] Michiel J. van Nieuwstadt and Richard M. Murray. Real time trajectory generation for differentially flat systems. To appear, *Int'l. J. Robust & Nonlinear Control*, 1996.