Motion Planning for Kinematic Stratified Systems With Application to Quasi-Static Legged Locomotion and Finger Gaiting

Bill Goodwine and Joel W. Burdick

***Abstract*—We present a general motion planning algorithm for robotic systems with a "stratified" configuration space. Such systems include quasi-static legged robots and kinematic models of object manipulation by finger repositioning. Our method is an extension of a nonlinear motion planning algorithm for smooth systems to the stratified case, where the relevant dynamics are not smooth. The method does not depend upon the number of legs or fingers; furthermore, it is not based on foot placement or finger placement concepts. Examples demonstrate the method.**

***Index Terms*—Stratified systems, nonlinear motion planning, legged locomotion, robotic manipulation.**

I. INTRODUCTION

THIS PAPER considers the motion planning problem for systems whose governing physics impose a "stratified" structure on the system's configuration space. A more formal notion of a stratified configuration space is presented in Section III. Stratification naturally arises in the context of legged locomotion and object manipulation via finger repositioning. These operations are characterized in part by the system making and breaking contact with its environment. The configuration spaces (or c-space) of these systems are "stratified" into subsets that correspond to different contact states. The governing dynamical equations depend upon the contact state, and are discontinuous during the making and breaking of contact.

The goal of our motion planning scheme is to determine the control inputs (e.g., mechanism joint variable trajectories) which will steer the walking robot from a starting configuration to a desired final configuration, or to manipulate the grasped object from an initial to a final configuration via a combination of finger repositioning and finger motions. The planner must simultaneously plan the mechanism's motion during a single contact state, as well as determine when to change contact states. This paper presents a general motion planning methodology for this class of systems, which includes all quasi-static legged locomotors and many kinematic models of multi-fingered hand manipulation. The method is independent of the number of legs (or fingers) and many other aspects of a robot's morphology. In the legged locomotion context, it is distinct from previous planning methods in that it is not based on foot placement concepts, and therefore the computationally burdensome calculation of foot placement can be avoided. Instead, our approach focuses on control inputs.

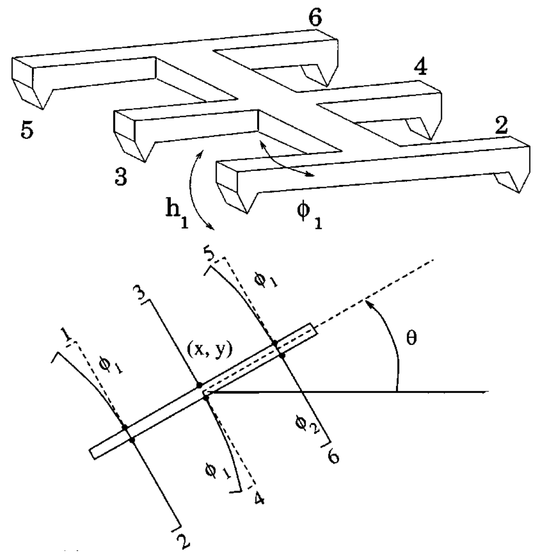

Fig. 1. Top: schematic of simple hexapod robot. Bottom: definition of kinematic variables.

As a concrete example of when such a planner is needed, consider the six-legged hexapod in Fig. 1 (this model will be fully explored in Section V). Each leg has only two degrees of freedom—the robot can only lift its legs up and down and move them forward and backward. Conventional hexapods are designed with three independent degrees of freedom per leg. The limited control authority in this design may be desirable in practical situations because it decreases the mechanical complexity of the robot. This leg geometry can also probably be implemented at very small size scales using MEMS technology. However, such decreased kinematic complexity comes at the cost of requiring more sophisticated control and motion planning theory. Note that for this model, it is not immediately clear if the robot can move "sideways."

The issue of this mechanism's ability to move sideways is the *controllability problem*. Previous work by the authors has considered controllability tests for stratified systems. We assume throughout this paper that a given system is *gait controllable* in

Manuscript received August 24, 2000; revised September 17, 2001. This paper was recommended for publication by Associate Editor P. Allen and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported in part by a grant from the Office of Naval Research.

B. Goodwine is with the Aerospace and Mechanical Engineering Department, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: goodwine@controls.ame.nd.edu).

J. W. Burdick is with the Engineering and Applied Science Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: jwb@robotics.caltech.edu).

Publisher Item Identifier S 1042-296X(02)02654-X.

the stratified sense as defined in [1] and [2]. Otherwise, it is not necessarily possible to determine a set of system inputs which steer the robot to the desired final configuration. Given the assumption of controllability, this paper addresses how to plan the robot's leg (or finger) movements so that it can approximately follow a given trajectory. A conventional "foot-placement" approach, where the foot can be placed as necessary to implement vehicle motion will clearly not work for the hexapod of Fig. 1, because sideways leg placement is impossible.

Our approach is an extension of the method by Lafferriere and Sussmann [3] for motion planning for a class of nonlinear kinematic systems whose equations of motion are smooth. However, since legged robots (and grasping hands) intermittently make and break contact, their equations of motion are not smooth. Hence, the method of [3] can not be directly applied. Section III introduces the notion of a *stratified* configuration space, which is decomposed into various subspaces (or strata) depending upon which combination of feet are in contact with the ground. We extend the approach of [3] by using the stratified c-space structure in a novel way. *It is likely that other methods for steering smooth systems (such as* [4]*) can be similarly extended by adopting our framework*. A main contribution of this work is the introduction of a geometric framework that supports the extension of prior nonholonomic motion planning techniques to this class of systems.

Our approach is general and thus works independently of the number of legs (fingers). It may be true that for a given quasi-static legged robot, one could develop a specific motion planner that would perform as well, or possibly better, than the technique described in this paper. The key advantage of this approach is its generality. It is particularly well suited to the task of quickly designing a planner during the preliminary stages of legged robot system design. While the techniques outlined in this paper are applicable to both locomotion and a class of hand manipulation problems, the bulk of the paper will focus on locomotion, with the application to hand manipulation briefly sketched at the end of the paper.

There is a *vast* literature on the analysis and control of legged robotic locomotion. Prior efforts have typically focused either on a particular morphology (e.g., biped [5], quadruped [6], [7], or hexaped [8]) or a particular locomotion assumption (e.g., quasi-static [8] or hopping [9]). Less effort has been devoted to uncovering principles that span all morphologies and assumptions. Some general results do exist. For example, the bifurcation analysis in [10], many optimal control results such as those in [11] and the fundamental conservation of momentum and energy results that underlie Raibert's hopping results [9] have general applicability. However, none of these methods directly use the inherent geometry of stratified configuration spaces to formulate results which span morphologies and assumptions. 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 toward establishing a solid basis for locomotion engineering.

In contrast to robotic legged locomotion, many results in robotic grasping and manipulation are formulated in a manner that is independent of the morphology of the gripper [12]. Vast efforts have been directed toward the *analysis* of grasp stability and force closure [13]–[15], motion planning assuming continuous contact [16]–[18] and haptic interfaces and other sensing [19]–[21]. *Finger gaiting*, where fingers make and break contact with the object has been less extensively considered. Finger gaiting has been implemented in certain instances [22]–[24] and also partially considered theoretically [25]–[27]. Perhaps the approach which most closely mirrors that of the subject of this proposal is in [12] where notions of controllability and observability from "standard" control theory are applied to grasping (however, these results are limited to the linear case and do not allow for fingers to intermittently contact the object). When applied to manipulation, our method can seemlessly integrate point contact finger repositioning, and is the first algorithm to do so with this amount of generality. Extending this method to include rolling contact manipulation is feasible, though nontrivial. This extension will be the subject of a future publication.

To make the paper self-contained, Section II presents some basic nonlinear control concepts and provides a brief overview of the motion planning method of [3] (due to space limitations, a comprehensive summary is, unfortunately, not possible). Section III introduces our notion of a stratified c-space. Section IV presents our algorithm in the context of quasi-static legged locomotion, while Section V applies this algorithm to the system of Fig. 1. Section VI sketches the application of these ideas to multi-fingered hand manipulation, and presents an example.

II. Background

We assume the reader is familiar with the basic notation and formalism of differential geometry and nonlinear control theory, as in [28]. The following definitions and classical theorems are reviewed so that the starting point of our development will be clear.

The equations of motion for smooth kinematic nonholonomic systems take the form of a driftless nonlinear affine control system evolving on a configuration manifold, M:

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

Since we restrict our analysis to quasi-static locomotion and kinematic models of multi-fingered manipulation, the governing equations of motion will piecewise take the form of (1) on each strata. 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

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

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

The results from [3] are formulated primarily by way of *formal computations*. For example, the flow along the vector field g_i can be considered by its *formal exponential* of g_i, denoted by

$$\phi_t^{g_i}(x) := e^{tg_i}(x) = \left(I + tg_i + \frac{t^2}{2}g_i^2 + \cdots\right) \tag{3}$$

where terms of the form g_i^k are partial differential operators (*not* vector fields) and Lie brackets are represented by commutation: $[g_1, g_2] = g_1g_2 - g_2g_1$. In order to use (3), composition must be from left to right, as opposed to right to left for flows, e.g., $\phi_{t_2}^{g_2} \circ \phi_{t_1}^{g_1} = e^{g_1t_1}e^{g_2t_2}$, where both sides of this equation mean "flow along g_1 for time t_1 and then flow along g_2 for time t_2." A rigorous justification for the use of such a formal representation is somewhat lengthy and can be found in [3], [29]. Essentially, the approach in [3] is to represent vector fields with "indeterminates," for which expansions of the form of (3) can be justified. The relationship between the flow along vector fields sequentially is given by the Campbell–Baker–Hausdorff formula [30].

Theorem 1: Given two smooth vector fields g_1, g_2 the composition of their exponentials is given by

$$e^{g_1}e^{g_2} = e^{g_1+g_2+\frac{1}{2}[g_1,g_2]+\frac{1}{12}([g_1,[g_1,g_2]]-[g_2,[g_1,g_2]])\cdots} \tag{4}$$

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

A. Trajectory Generation for Smooth Systems

A nonholonomic control system typically does not have enough controls to directly drive the system along a given trajectory, i.e., the number m in (1) is less than the c-space dimension. In the method of [3], this deficit is managed by using an "extended system," where "fictitious controls," corresponding to higher order Lie bracket motions, are added. If enough Lie brackets are added to the system to span all possible motion directions (which is possible if the system is locally controllable), then the motion planning problem becomes trivial for the extended system.

The *extended system* is constructed by adding Lie brackets to the original system from (1),

$$\dot{x} = b_1v_1 + \cdots b_mv_m + b_{m+1}v_{m+1} + \cdots + b_sv_s \tag{5}$$

where $b_i = g_i$ for $i = 1, \ldots, m$, and the $b_{m+1}, \ldots, b_s$ correspond to higher order Lie brackets of the g_i, chosen so that $\dim(\mathrm{span}\{b_1(x), \ldots, b_s(x)\}) = \dim(T_xM)$ at each $x \in M$. The v_i's where $i > m$ are called *fictitious inputs* since they do not correspond with any actual system inputs. A technical requirement is that the higher order Lie brackets must belong to the Philip Hall basis (see [31], [32] for a definition) for the Lie algebra of vector fields on M.

The control inputs v_i which steer the extended system can be found as follows. If it is desired to start at the point p and finish at the point q, define a curve $\gamma(t), t \in [0, 1]$ connecting p and q and solve

$$\dot{\gamma}(t) = b_1(\gamma(t))v_1 + \cdots + b_s(\gamma(t))v_s \tag{6}$$

for the fictitious controls v_i. This will simply involve inverting a square matrix or determining a pseudo-inverse, depending on whether or not there are more b_i's than the dimension of the configuration space, i.e., whether $s > m$.

A basic fact from differential geometry is that all flows of (1) can be represented in the (formal) form

$$S(t) = e^{h_s(t)b_s}e^{h_{s-1}(t)b_{s-1}} \ldots e^{h_2(t)b_2}e^{h_1(t)b_1} \tag{7}$$

for some functions $h_1, h_2, \ldots, h_s$, called the *backward Philip Hall coordinates*. Furthermore, as shown in [3], $S(t)$ satisfies the formal differential equation

$$\dot{S}(t) = S(t)(b_1v_1 + \cdots + b_sv_s), \qquad S(0) = 1. \tag{8}$$

If we define the *adjoint mapping*

$$\mathrm{Ad}_{e^{-h_ib_i}}b_j = e^{-h_ib_i}b_je^{h_ib_i},$$

then it is straightforward to show that

$$\mathrm{Ad}_{e^{-h_ib_i}\ldots e^{-h_{j-1}b_{j-1}}}b_j\dot{h}_j = \left(\sum_{k=1}^{s} p_{j,k}(h)b_k\right)\dot{h}_j, \tag{9}$$

for some polynomials $p_{j,k}(h)$. (For a complete derivation, see [32]). Equating coefficients of the b_i in (8) with the derivative of (7), and using (9), yields differential equations having the form

$$\dot{h} = A(h)v \quad h(0) = 0. \tag{10}$$

These equations specify the evolution of the backward Philip Hall coordinates in response to the fictitious inputs, which were found via (6).

Example 1: Here we present a simple example illustrating the computation of the Philip Hall coordinates. More complicated examples are presented in detail in [32]. Consider a two input system on a three dimensional configuration manifold:

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

Assume that the set $\{g_1, g_2, [g_1, g_2]\}$ spans T_xM at all $x \in M$. The extended system is

$$\dot{x} = b_1(x)v_1 + b_2(x)v_2 + b_3(x)v_3,$$

where

$$b_1 = g_1$$
$$b_2 = g_2$$
$$b_3 = [g_1, g_2].$$

The formal differential equation is

$$\dot{S}(t) = S(t)(b_1v_1 + b_2v_2 + b_3v_3), \tag{11}$$

and the formal exponential is

$$S(t) = e^{h_3(t)b_3}e^{h_2(t)b_2}e^{h_1(t)b_1}.$$

Differentiating the formal exponential yields:

$$\dot{S}(t) = e^{h_3(t)b_3}\dot{h}_3(t)b_3e^{h_2(t)b_2}e^{h_1(t)b_1} + e^{h_3(t)b_3}e^{h_2(t)b_2} \times \dot{h}_2(t)b_2e^{h_1(t)b_1} + e^{h_3(t)b_3}e^{h_2(t)b_2}e^{h_1(t)b_1}\dot{h}_1(t)b_1,$$

rearranging gives

$$\dot{S}(t) = S(t)\left(e^{-h_1(t)b_1}e^{-h_2(t)b_2}\dot{h}_3(t)b_3e^{h_2(t)b_2}e^{h_1(t)b_1}\right) + S(t)\left(e^{-h_1(t)b_1}\dot{h}_2(t)b_2e^{h_1(t)b_1}\right) + S(t)\dot{h}_1(t)b_1,$$

and using the adjoint notation

$$\dot{S}(t) = S(t)\left(\mathrm{Ad}_{e^{-h_2(t)b_2}e^{-h_1(t)b_1}}\dot{h}_3(t)b_3 + \mathrm{Ad}_{e^{-h_1(t)b_1}}\dot{h}_2(t)b_2 + \dot{h}_1(t)b_1\right), \quad (12)$$

which is in the form of (8).

Expanding the formal exponentials to second order according to (3) gives the following coefficients for each $\dot{h}_i(t)$ in the preceding equation:

$$\begin{aligned}&\dot{h}_1(t): b_1\\&\dot{h}_2(t): b_2 - h_1(t)(b_1b_2 - b_2b_1) = b_2 - h_1(t)b_3\\&\dot{h}_3(t): b_3.\end{aligned}$$

In particular, the coefficient of $\dot{h}_2(t)$ is computed as follows (terms higher than second order are dropped):

$$\begin{aligned}&\mathrm{Ad}_{e^{-h_1(t)b_1}}\dot{h}_2(t)b_2\\&\quad= \left(I - h_1(t)b_1 + \frac{1}{2}h_1^2(t)b_1^2\right)\\&\qquad\times \dot{h}_2(t)b_2\left(I + h_1(t)b_1 + \frac{1}{2}h_1^2(t)b_1^2\right)\\&\quad= (b_2 - h_1(t)b_1b_2)\left(I + h_1(t)b_1 + \frac{1}{2}h_1^2(t)b_1^2\right)\dot{h}_2(t)\\&\quad= (b_2 + h_1(t)b_2b_1 - h_1(t)b_1b_2)\dot{h}_2(t)\\&\quad= (b_2 - h_1(t)[b_1, b_2])\dot{h}_2(t)\\&\quad= (b_2 - h_1(t)b_3)\dot{h}_2(t).\end{aligned}$$

Equating the coefficients of the b_i's in (12) and (11) (using the preceding computations)

$$\begin{aligned}\dot{h}_1(t) &= v_1\\\dot{h}_2(t) &= v_2\\\dot{h}_3(t) &= h_1(t)v_2 + v_3.\end{aligned}$$

Solving these differential equations for the h_i provides the amount of time the system must flow along each of the extended system basis vector fields in (7). □

Next one must determine the actual inputs using the Philip Hall coordinates. For systems with extended systems only including Lie brackets up to second order, this is a straightforward procedure. Having determined the Philip Hall coordinates from (10), the flow for the system is of the form of (7), where the Philip Hall coordinates, $h_i(t)$ are now known. Therefore, the final position of the system is given by

$$S(1) = e^{h_s(1)b_s}e^{h_{s-1}(1)b_{s-1}}\cdots e^{h_2(1)b_2}e^{h_1(1)b_1}$$

(recall that for the formal representation, composition is from left to right). Since the exponentials $e^{h_s(1)b_s}, e^{h_{s-1}(1)b_{s-1}}, \ldots, e^{h_{m+1}(1)b_{m+1}}$ are second order Lie brackets, each of the individual flows starting from the left-most exponential can be simply represented by a sequence of four piecewise constant inputs as indicated in (2). The exponentials $e^{h_m(1)b_m}, e^{h_{m-1}(1)b_{m-1}}, \ldots, e^{h_1(1)b_1}$ simply represent concatenated flows directly along each of the control vector fields $g_1(x), g_2(x), \ldots, g_m(x)$, these flows are accomplished by letting the corresponding input be "on" (i.e., $u_i = 1$) for the corresponding time represented by $h_i(1)$.

We only provide detailed computations for systems of degree two since, in practice physical systems that require Lie bracket motions of order greater than two may be inconvenient to control since many motions are needed to effect even a small motion in a higher order Lie bracket direction.

Example 2: Returning to the previous example, since $h_i(t), i = 1, 2, 3$ are known, the final sequence of flows is given by

$$\underbrace{u_1(\sqrt{h_3(1)}) \circ u_2(\sqrt{h_3(1)}) \circ -u_1(\sqrt{h_3(1)}) \circ -u_2(\sqrt{h_3(1)})}_{\text{Lie bracket } b_3=[g_1,g_2] \text{ approximation}}$$

followed by

$$u_2(h_2(1)) \circ u_1(h_1(1))$$

where the notation $u_i(t)$ means that $u_i = 1$ for time t if $t \geq 0$ or $u_i = -1$ for time $|t|$ if $t < 0$ and $\circ$ means concatenation, i.e., $u_1(t) \circ u_2(t)$ means that the flow for u_2 follows the flow for u_1. □

For systems with Lie brackets of order greater than two in the extended system, the procedure involves some additional steps to which we direct the interested reader to [3], [29]. In particular, for higher order systems, 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 an algebraic transformation (see [3]). Additionally, since the piecewise approximation to the flow along Lie bracket is only approximate, relatively straightforward corrections to this must be computed when determining a piecewise approximation to flows along vector fields of degree greater than two.

If the system is nilpotent,[1] this method exactly steers the system to the desired final state; otherwise, the system is steered to a point that is, at worst, half the distance to the desired state [3]. The algorithm can be iterated to generate arbitrary precision. This iterated method also includes the notion of a "critical" step length. Reference [3] estimates the critical step length bound, and shows via simulations that the actual critical length is typically larger than the estimated bound.

III. Stratified Configuration Spaces

The method reviewed in Section II-A can not be directly used for legged or multi-fingered robots because their governing equations of motion are not smooth. To adapt this method (and similar nonholonomic motion planning methods) to these systems, we use the notion of a *stratified* configuration space. While the stratified concept is equally applicable to locomotion

[1]A system of the form (1) is said to be *nilpotent of order* k if all the Lie brackets between control vector fields of order greater than k are 0.

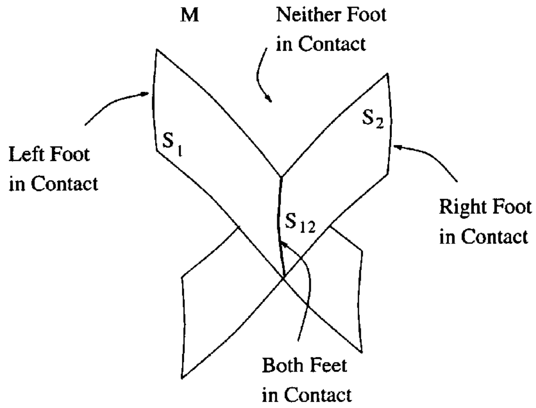

Fig. 2. Abstract depiction of the stratified structure of a biped robot c-space.

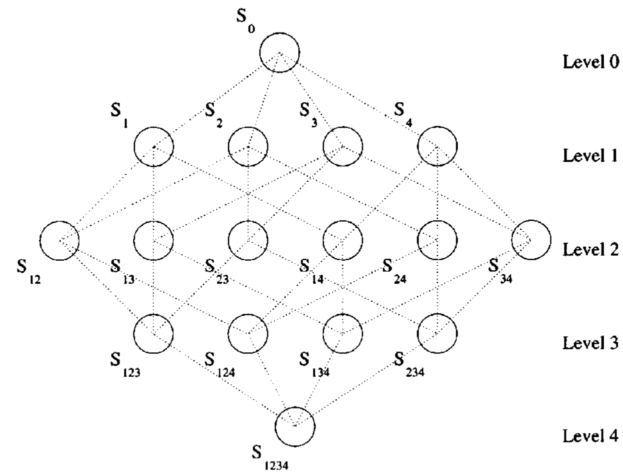

Fig. 3. Four-level stratification.

and multi-fingered manipulation, the language of locomotion is used below for simplicity.

Let S_0 denote a robot's configuration manifold, which describes the robot's position and orientation as well as all of the mechanism's joint variables. The robot's possible configurations will be subjected to constraints if one or more of its feet (fingers) are in contact with the ground (object). The set of configurations corresponding to one contact is generically a codimension one submanifold of S_0. Let $S_i \subset S_0$ denote the codimension one submanifold of S_0 that corresponds to all configurations where only the ith foot contacts the terrain. That the $\{S_i\}$ are submanifolds can be demonstrated by noting that set of points corresponding to ground contact can be described by the preimage of a function describing the foot's height. We generally assume that S_i, is, at least locally, defined by a level set of a function $\Phi_i(x){:}S_0 \to \mathbb{R}$. For legged robotic locomotion systems, these functions, Φ_i are naturally defined by the height of the robot's foot off of the ground so that the level sets $\Phi_i^{-1}(0)$ are of interest.

When both the ith and jth feet are on the ground, the corresponding set of states is a codimension 2 submanifold of S_0 that is formed by the intersection of the two single contact submanifolds. Denote, the intersection of S_i and S_j, by $S_{ij} = S_i \cap S_j$. The structure of the configuration manifold for a biped is abstractly illustrated in Fig. 2. For systems with larger numbers of legs (fingers), further intersections, corresponding to more complicated contact states, can be similarly defined in a recursive fashion: $S_{ijk} = S_i \cap S_j \cap S_k = S_i \cap S_{jk}$, etc. Denote an arbitrary intersection set (or "stratum") by $S_I = S_{i_1 i_2 \ldots i_n}, I = \{i_1 i_2 \ldots i_n\}$, and assume that S_I is a regular submanifold of S_0. This is generically true for rigid body mechanisms. If the strata $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ are locally described by the functions $\Phi_{i_1}, \Phi_{i_2}, \ldots, \Phi_{i_k}$, then S_I will be a submanifold of S_0 if the functions $\Phi_{i_1}, \Phi_{i_2}, \ldots, \Phi_{i_k}$ are functionally independent. If the functions Φ_I correspond to foot heights, this functional independence will be satisfied for legged robots.

We say that the robot c-space is *stratified*[2] and call each of the submanifolds S_I a *stratum*. The highest codimension stratum containing the point x is called the *bottom stratum*, and any other submanifolds containing x are called *higher strata*. When making comparisons among different strata, we will refer to higher codimension (i.e., lower dimensional) strata as *lower strata*, and lower codimension (i.e., higher dimensional) strata 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, some contacts are better characterized by nonholonomic constraints. Regardless of the constraint type, the system's equations of motion will change in a nonsmooth manner. Otherwise, the system's equations of motion are smooth, though generally different in each strata. Hence, the discontinuities are localized to regions of transition between strata.

The equations of motion at $x \in S_I$ are written as

$$\dot{x} = g_{I,1}(x)u_{I,1} + \cdots g_{I,n_I}(x)u_{I,n_I} \tag{13}$$

where n_I depends upon the codimension of S_I and the nature of the additional constraints imposed on the system in S_I. We assume that the vector fields in the equations of motion for any given stratum are well defined at all points in that stratum, including points contained in any substrata of that stratum. For example, the vector fields $g_{0,i}(x)$ are well defined for $x \in S_i$. Note, however, that they do *not* represent the system's equations of motion in the substrata, but, nonetheless, are still well defined as vector fields.

Fig. 3 illustrates, via a graph-like structure, a four-level stratification, which corresponds to a four-legged walker. A node corresponds to a stratum, and the presence of an edge connecting nodes indicates that it is possible to move between the strata that are connected by the edge. The ability to move between two strata depends upon the mechanics of a given problem, and will generally be obvious from the characteristics of a given problem. Whether or not edges between nodes are permissible is considered in more detail in [2]. When a configuration manifold is consistent with the above description, we will refer to it as a *stratified configuration space*.

Definition 1: Let S_0 be a manifold, and n functions $\Phi_i{:}S_0 \mapsto \mathbb{R}, i = 1, \ldots, n$ be such that the level sets $S_i = \Phi_i^{-1}(0) \subset S_0$ are regular submanifolds of S_0, for each i, and the intersection

[2]Note that the terms "stratification" and "strata" are also used in other contexts to describe the topology of orbit spaces of Lie group actions, and are a slight generalization of the notion of a foliation [33].

of any number of level sets, $S_{i_1 i_2 \ldots i_m} = \Phi_{i_1}^{-1}(0) \cap \Phi_{i_2}^{-1}(0) \cap \cdots \cap \Phi_{i_m}^{-1}(0), m \leq n$, is also a regular submanifold of S_0. Then S_0 and the functions Φ_n define a *stratified configuration space*.

Locomotion gaits have a straightforward interpretation in a stratified configuration space. In particular, we specify a *gait* as an ordered sequence of strata:

$$\mathcal{G} = \{S_{I_1}, S_{I_2}, \ldots, S_{I_n}, S_{I_{n+1}} = S_{I_1}\} \tag{14}$$

where n is the number of different contact states in the gait. In this ordered sequence, the first and last element are identical, indicating that the gait is a closed loop in the strata graph. For the gait to be meaningful, the system must be able to switch from stratum S_{I_i} to $S_{I_{i+1}}$ for each i. We further assume that the specified gait or gaits satisfy the gait controllability conditions of [2] so that arbitrary trajectories can be tracked.

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

$$\Delta_{S_I} = \text{span}\{g_{S_{I,1}}, \ldots, g_{S_{I,n_I}}\}.$$

The *involutive closure* of Δ_{S_I}, denoted by $\bar{\Delta}_{S_I}$, is the closure of Δ_{S_I} under Lie bracketing. The controllability of a given gait, (14), can be determined by letting $\mathcal{D}_1 = \bar{\Delta}_{I_1}$. If $S_{I_{i-1}} \subset S_{I_i}$, then $\mathcal{D}_i = \mathcal{D}_{i-1} + \bar{\Delta}_{I_i}$. Else, if $S_{I_i} \subset S_{I_{i-1}}$, then $\mathcal{D}_i = (\mathcal{D}_{i-1} \cap TS_{I_i}) + \bar{\Delta}_{I_i}$ In [2] it s shown that if $\dim(\mathcal{D}_n) = \dim(T_{x_0} S_B)$ the system is *gait controllable* from x_0. For a more rigorous discussion and summary of stratified system controllability, see [1] and [2].

IV. Legged Trajectory Generation

This section extends the procedure outlined in Section II to kinematic systems having a stratified c-space. We focus on quasi-static legged locomotion in this section. Section VI sketches the extension to basic finger gaiting manipulation.

Assume that the robot starts at a configuration p and seeks to reach a final configuration q. By a configuration, we mean the position and orientation of the body, as well as the configuration of the legs. We assume that both p and q lie in the same bottom stratum, denoted by S_B. This corresponds to the legged robot starting and stopping with the same set of feet in contact with the ground. Eliminating this requirement is a simple extension of the algorithm described below.

The switching behavior associated with stratified systems can not be accounted for in the methods of Section II.A. However, the method can be extended to legged and fingered robotic systems via the notion of a *stratified extended system* on S_B.

A. The Stratified Extended System

On each strata, only one set of controls (or governing equations) is in effect. Generally, the equations of motion in the bottom strata will be different than those in higher strata. Furthermore, it will be typically true that the goal q can not be reached by remaining in S_B. Hence, some switching amongst the strata will be necessary. 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. As shown below, it is possible to consider the vector fields associated with each stratum in one common space. In this case, that common space will be the bottom stratum. This concept will be encapsulated below in the definition of a "stratified extended system." We first introduce some examples to show how we can consider vector fields defined on different strata in a common space. Additional examples that deal with more subtle issues can be found in [34].

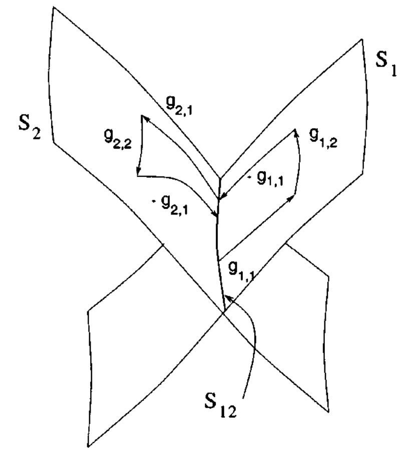

Fig. 4. Sequence of flows.

Example 3: Consider the conceptual biped configuration space as shown in Fig. 2. Assume that on stratum S_{12}, the vector field $g_{1,1}$ moves the system off of S_{12} and onto S_1, and correspondingly, $g_{2,1}$ moves the system off of S_{12} onto S_2. Also, we consider the vector fields $g_{1,2}$ and $g_{2,2}$, defined on S_1 and S_2 respectively. Consider the following sequence of flows, starting from the point $x_0 \in S_{12}$

$$x_f = \underbrace{\phi_{-g_{2,1}}^{t_6}}_{S_{12} \leftarrow S_2} \circ \underbrace{\phi_{g_{2,2}}^{t_5}}_{\text{on } S_2} \circ \underbrace{\phi_{g_{2,1}}^{t_4}}_{S_2 \leftarrow S_{12}} \circ \underbrace{\phi_{-g_{1,1}}^{t_3}}_{S_{12} \leftarrow S_1} \circ \underbrace{\phi_{g_{1,2}}^{t_2}}_{\text{on } S_1} \circ \underbrace{\phi_{g_{1,1}}^{t_1}}_{S_1 \leftarrow S_{12}} (x_0). \tag{15}$$

The notation "$S_{12} \leftarrow S_1$" means that the flow takes the system from S_1 to S_{12} and "on S_1" means that the flow lies entirely in S_1. This sequence of flows is illustrated in Fig. 4. In this sequence, the system first moved off of the bottom stratum into S_1, flowed along the vector field $g_{1,2}$, flowed back onto the bottom stratum, off of the bottom stratum onto S_2, along vector field $g_{2,2}$ and back to the bottom stratum.

Notice that from the Campbell–Baker–Hausdorff formula [see (4)], if the Lie bracket between two vector fields is zero, then their flows commute. Thus, if

$$[g_{1,1}, g_{1,2}] = 0 \quad \text{and} \quad [g_{2,1}, g_{2,2}] = 0, \tag{16}$$

we can reorder the sequence of flows in (15) by interchanging the flow along $g_{1,1}$ and $g_{1,2}$ and the flows along $g_{2,1}$ and $g_{2,2}$ as follows

$$\begin{aligned} x_f &= \underbrace{\phi_{g_{2,2}}^{t_5} \circ \phi_{-g_{2,1}}^{t_6}}_{\text{interchanged}} \circ \phi_{g_{2,1}}^{t_4} \circ \underbrace{\phi_{g_{1,2}}^{t_2} \circ \phi_{-g_{1,1}}^{t_3}}_{\text{interchanged}} \circ \phi_{g_{1,1}}^{t_1}(x_0) \\ &= \underbrace{\phi_{g_{2,2}}^{t_4} \circ \phi_{g_{1,2}}^{t_2}}_{\text{on } S_{12}}(x_0), \end{aligned} \tag{17}$$

if $t_1 = t_3$ and $t_4 = t_6$. □

Note that $g_{1,2}$ and $g_{2,2}$ are vector fields in the equations of motion for strata S_1 and S_2, respectively, but *not* on stratum S_{12}. However, the sequence of flows in (15) occurs on different strata, where the flows are governed by vectors fields associated with each stratum. This flow yields the same net result as the net flow in (17), where the vector fields are evaluated on the bottom stratum, even though they are not part of the equations of motion there. Furthermore, we note that if the vector fields $g_{1,2}$ and $g_{2,2}$ are tangent to the substratum S_{12}, then the resulting flow given in (17) will remain in S_{12}. In fact, it is implicitly required in the above argument that at least $g_{1,2}$ is tangent to S_{12}.

If the bottom stratum is described by the level set of a function Φ_B, and if a vector field $g_{1,2}$ is not tangent to the bottom stratum, then $\langle \mathbf{d}\Phi_B, g_{1,2}\rangle = f_1 \neq 0$. Also, since the vector field $g_{1,1}$ moves the foot out of contact, we similarly have $\langle \mathbf{d}\Phi_B, g_{1,1}\rangle = f_2 \neq 0$. Then, the vector field, $\tilde{g}_{1,2} = g_{1,2} - (f_1/f_2)g_{1,1}$, is tangent to S_B because

$$\langle \mathbf{d}\Phi_B, \tilde{g}_{1,2}\rangle = \langle \mathbf{d}\Phi_B, g_{1,2}\rangle - \frac{f_1}{f_2}\langle \mathbf{d}\Phi_B, g_{1,1}\rangle = 0. \qquad (18)$$

Henceforth, we will just assume that the vector field on the higher stratum is tangent to the lower stratum, and note that if it is not tangent, we can modify it to be so in the above manner.

The above example shows how one can effectively determine the influence of a control that is defined in a higher stratum on the net evolution of the system in the lower stratum. The following example shows how motions that are analogous to Lie Bracket motions can be realized by controls on *different* higher strata.

Example 4: Consider the sequence of flows

$$\begin{aligned} x_f = \phi^{t_{12}}_{-g_{2,1}} \circ \phi^{t_{11}}_{-g_{2,2}} \circ \phi^{t_{10}}_{g_{2,1}} \circ \phi^{t_9}_{-g_{1,1}} \circ \phi^{t_8}_{-g_{1,2}} \circ \phi^{t_7}_{g_{1,1}} \\ \circ \phi^{t_6}_{-g_{2,1}} \circ \phi^{t_5}_{g_{2,2}} \circ \phi^{t_4}_{g_{2,1}} \circ \phi^{t_3}_{-g_{1,1}} \circ \phi^{t_2}_{g_{1,2}} \circ \phi^{t_1}_{g_{1,1}}(x_0) \end{aligned}$$

The first six flows in this example are the same as in Example 3. Following the first six flows are six more wherein the flows that are entirely on S_1, i.e., the flow along $g_{1,2}$, and entirely on S_2, i.e., the flow along $g_{2,2}$, are in the negative direction. If the Lie brackets are zero as in (16), and $t_i = t_{i+2}, i = 1,4,7,10$ these flows can be rearranged as

$$x_f = \phi^{t_{11}}_{-g_{2,2}} \circ \phi^{t_8}_{-g_{1,2}} \circ \phi^{t_5}_{g_{2,2}} \circ \phi^{t_2}_{g_{1,2}}(x_0).$$

Now, if $t_2 = t_5 = t_8 = t_{11}$

$$\begin{aligned} x_f &= \phi^{t_{11}}_{-g_{2,2}} \circ \phi^{t_8}_{-g_{1,2}} \circ \phi^{t_5}_{g_{2,2}} \circ \phi^{t_2}_{g_{1,2}}(x_0) \\ &= \phi^{t^2}_{[g_{1,2}, g_{2,2}]} + \mathcal{O}(t^3)(x_0), \end{aligned}$$

where $t = t_2 = t_5 = t_8 = t_{11} \ll 1$. Thus, this sequence provides a net flow in S_{12} in the direction of the Lie bracket between vector fields which are in the equations of motion on different strata, S_1 and S_2. □

In Examples 3 and 4, it was required that certain Lie brackets be zero. While one could simply check that these conditions are met in a given situation, the following assumption will guarantee this condition.

Assumption 1: If it is necessary to lift a foot from the ground during a gait cycle, we assume that the robot can directly control, (via a single control, or a combination of control inputs), the height of that foot relative to the ground. Furthermore, for each stratum comprising the given gait, we assume that the system's equations of motion are independent of the foot height, i.e., the robot's motion is independent of whether a particular foot is very close to the ground, or very far from the ground, but may be dependent upon whether or not a foot is in or out of contact with the ground. When this is so, the Lie bracket of the vector field controlling foot height with any other vector field is zero, and the decoupling requirement is satisfied. Additionally, the tangency requirements for canceling the flows associated with raising and lowering the foot will automatically be satisfied.

This is arguably a strict assumption. However, for kinematic, legged robots this assumption will almost always be satisfied (see Section V for an example).

Examples 3 and 4 show that in given a stratified system, the vector fields on any stratum (other than vector fields corresponding to lifting or replacing feet) can be considered as part of the equations of motion in the bottom stratum if either certain Lie bracket and tangency conditions are met, or if Assumption 1 is satisfied. If the vector fields are not tangent to the bottom stratum, they are modified as in Example 3.

We have shown above that it is possible to consider vector fields in higher strata as part of the equations of motion for the system on the bottom stratum. Based on this observation, we introduce the following.

Definition 2: Extended Stratified System The *extended stratified system* on the bottom strata, S_B, is the driftless system comprised of the vector fields on the bottom strata, chosen vector fields from the higher strata, and Lie brackets of vector fields from S_B and higher strata, i.e., it is a system taking the form:

$$\begin{aligned} \dot{x} = b_1(x)v_1 + \cdots b_m(x)v_m + \underbrace{b_{m+1}v_{m+1}\cdots + b_nv_n}_{\text{from higher strata}} \\ + \underbrace{b_{n+1}v_{n+1} + \cdots + b_pv_p}_{\text{any Lie brackets}} \end{aligned} \qquad (19)$$

where the $\{b_1, \ldots, b_p\}$ span T_xS_0, the inputs $v_1, \ldots, v_n$ are real, and the inputs $v_{n+1}, \ldots, v_p$ are fictitious.

With this definition, we have effectively increased the class of vector fields that we may employ when using the motion planning algorithm presented in Section II.

B. The Motion Planning Algorithm

For the purposes of motion planning, the method presented in Section II could be used in conjunction with the stratified extended systems. The basic idea is to use the stratified extended system to plan the motion in the bottom stratum in order to obtain the fictitious inputs. We can determine the actual inputs by the method in Section II with the modification that whenever the system must flow along a vector field in a higher stratum, it switches to that stratum by lifting the appropriate foot or feet, flowing along the vector field, and then replacing the appropriate foot or feet, as in Example 3.

Specifically, the algorithm to generate trajectories that move the system from initial configuration p to final configuration q is as follows.

1) Construct the *extended stratified system*, (19), on the bottom strata S_B.
2) Find a nominal trajectory, $\gamma(t)$, that connects p and q. Given $\gamma(t)$, solve

$$\dot{\gamma}(t) = b_1(x)v_1 + \cdots + b_p(x)v_p$$

for the fictitious inputs, v_i. As discussed in Section IV.C, it may be necessary to decompose the entire trajectory from the initial point to final point into smaller subtrajectories.
3) For each path segment in each strata, compute the actual controls that steer the system along $\gamma(t)$. As discussed previously, this solution might require the transformation of the backward Philip Hall coordinates to forward Philip Hall coordinates if the degree of the Lie brackets in the extended system is greater than two.
4) Flow along each first order vector field, and approximate higher order vector fields as illustrated in Example 3. In general, it will be necessary to switch strata between some of these flows.

C. Gait Stability

Before we illustrate this method in Section V, we consider the additional issue of and stability. There is not an inherent mechanism in the straightforward application of the method of Section II to guarantee the stability of the gait. Recall that the method is based on the selection of a trajectory for the extended system, $\gamma(t)$, from which the fictitious inputs are determined. It is important to note that the actually realized trajectory will generally *not* be $\gamma(t)$. Thus, merely picking an initial trajectory $\gamma(t)$ which is always stable is not sufficient. One also must guarantee that the method's inherent deviations from the initial trajectory lie within the stability bounds.

Stability considerations can be incorporated into the method as follows. Assume that there is a means for determining the stability of the system by means of a scalar-valued function of the configuration $\Psi(x)$. For convenience, assume that when $\Psi(x) < 0$, the system is unstable, when $\Psi(x) > 0$, the system is stable, and when $\Psi(x) = 0$, the system is on the stability boundary. In our analysis, the initial trajectory, $\gamma(t)$, must be selected such $\Psi(\gamma(t)) > 0$.

The overall approach is to, when necessary, take steps that are "small enough" to ensure that the system remains stable. Since the flow sequences are composed of small motions and a norm is necessary to measure the length of a flow, we will either consider the system locally in $\mathbb{R}^n$ or equip the configuration manifold with a metric. Given a desired step along the trajectory, $\gamma(t), t \in [0, 1]$, let $\mathcal{R} = \min\{\|x - c\|, c \in \Psi^{-1}(0)\}$, i.e., the distance from the step's starting point to the closest point on the stability boundary. We want to ensure that the system's trajectory does not intersect the set $\Psi^{-1}(0)$. Let x_s and x_f denote the starting and final trajectory points. Without loss of generality, let $\gamma(t) = x + t(x_f - x_s)$ be a desired straight line path between the starting and end points. Also, let $\Delta = \|x_f - x_s\|$. Recall that the fictitious inputs, v_i were determined by solving an equation of the form $\dot{\gamma}(t) = g_1(\gamma(t))v_1 + \cdots + g_s(\gamma(t))v_s$ for the v_i. We have that $\|v_i\| < C\|\dot{\gamma}(t)\| = C\Delta$, for some constant C. By the method of constructing of the real inputs from the fictitious inputs, we have that $\|u_i\| < C\Delta^{1/k}$, where k is the degree of nilpotency of the system, or the degree of the nilpotent approximation.

Pick a ball, $\mathcal{B}$, of radius $\mathcal{R}$, and let K be the maximum norm of all the (first order) vector fields, g_i for all points in $\mathcal{B}$. Recall that the real inputs, u_i were given by a sequence of inputs which approximate the flow of the extended system. Denote this sequence by u_j^i, where the superscript indexes the input, and the subscript indexes its position in the sequence. The maximum distance that the system can possibly flow from the starting point, x_s, is given by the sum of the distances of the individual flows. Let $x_m = \max_{t\in[0,1]}\{\|x(t) - x_s\|\}$ denote the point in the flow that is maximally distant from the starting point (this is not necessarily the final point, x_f). To guarantee stability, we must show that $\|x_m - x\| < \mathcal{R}$. However, this distance, $\|x_m - x\|$ is necessarily bounded by the sum of the norms of each individual flow associated with one real control input, u_j^i, i.e.,

$$\|x_m - x\| \leq \sum_{i,j} \left\| \int_0^1 g_i u_j^i \, dt \right\|.$$

However, $\|u_j^i\| \leq C\Delta^{1/k}$ and $\|g_i(x)\| \leq K \;\forall x \in \mathcal{B}$. Thus

$$\|x_m - x\| \leq \sum_{i,j} KC\Delta^{1/k} \tag{20}$$

and since $\Delta = \|x_f - x\|$, by choosing the desired final point close enough to the starting point, the trajectory will not intersect the stability boundary.

Note that since Δ is raised to the power of $1/k$, if k is large, then it may be necessary to make Δ exceedingly small in order to ensure stability. However, the bound expressed in (20) is itself very conservative since it sums the length of a bound on each individual flow in the series. In actuality, because the largest flows correspond to the Lie brackets of order k, simply summing their component lengths will give a conservative bound. Given these two observations, an appropriate step length may often be best determined experimentally.

The same observations also apply to obstacle avoidance. If the robot traverses an environment with obstacles, we assume that the nominal trajectory is designed by an holonomic or rigid body planner in such a manner as to avoid obstacles. Ensuring that the actual trajectory also avoids the obstacles, requires that the nominal trajectory be analogously broken into sufficiently small steps to ensure that the actual trajectory remains sufficiently close to it.

V. Example

The approach is illustrated by generating control inputs that will steer a simplified model of the hexapod of Fig. 1 to walk over flat terrain (see Section VI for an example involving manipulation of a curved object, which is analogous to locomotion over uneven terrain). This hexapod model is adapted from a similar robot model presented previously in [35]. The key difficulty in this example is the fact that the legs are kinematically insufficient, making sideways motion difficult. Assume that the robot walks with a tripod gait[3], alternating movements of legs 1-4-5 with movements of legs 2-3-6. With the tripod gait assumption, this robot has four control inputs. The inputs u_1 and u_2 respectively control the forward and backward angular leg displacements of legs 1-4-5 and legs 2-3-6, while inputs u_3 and u_4 respectively control the height of legs 1-4-5 and 2-3-6.

The equations of motion can be written as follows.

$$\begin{aligned} \dot{x} &= \cos\theta(\alpha(h_1)u_1 + \beta(h_2)u_2) \\ \dot{y} &= \sin\theta(\alpha(h_1)u_1 + \beta(h_2)u_2) \\ \dot{\theta} &= l\alpha(h_1)u_1 - l\beta(h_2)u_2 \\ \dot{\phi}_1 &= u_1; \quad \dot{\phi}_2 = u_2 \\ \dot{h}_1 &= u_3; \quad \dot{h}_2 = u_4 \end{aligned}$$

where (x, y, θ) represents the planar position of the center of mass, ϕ_1 is the front to back angular deflection of legs 1-4-5, ϕ_2 is the angular deflection of legs 2-3-6, l is the leg length and h_i is the height of the legs off the ground. The functions $\alpha(h_1)$ and $\beta(h_2)$ are defined by

$$\alpha(h_1) = \begin{cases} 1, & \text{if } h_1 = 0 \\ 0, & \text{if } h_1 > 0 \end{cases} \qquad \beta(h_2) = \begin{cases} 1, & \text{if } h_2 = 0 \\ 0, & \text{if } h_2 > 0 \end{cases}.$$

Note that these equations require some foot slippage in order to describe the motion of a robot like the one illustrated in Fig. 1. Since the robot walks in a tripod gait, stability is ensured if the robot's center of mass remains above the triangle defined by the tripod of feet which are in contact with the ground. Considering the motion of legs 1-4-5, the center of mass of the robot must be at least $b = (l_b/4) + l\sin\phi_1$ from the front of the robot to ensure stability, where l_b denotes the length of the body. See Fig. 5. Alternatively, if the center of mass is located a distance b from the front of the robot, then stability is ensured during the motion if both of these constraints are satisfied

$$\begin{aligned} \phi_1 &< \sin^{-1}\left(\left(b - \frac{l_b}{4}\right)\Big/ l\right) \\ \phi_2 &> -\sin^{-1}\left(\left(\frac{3l_b}{4} - b\right)\Big/ l\right). \end{aligned}$$

Denote the stratum when all the feet are in contact ($\alpha = \beta = 1$) by S_{12}, the stratum when tripod one is in contact ($\alpha = 1, \beta = 0$), by S_1, the stratum when tripod two is in contact ($\alpha = 0, \beta = 1$), by S_2 and the stratum when no legs are in contact ($\alpha = \beta = 0$), by S_0. Note that this system satisfies the requirements of Assumption 1 since, regardless of the values of α and β, the vector fields moving the foot out of contact with the ground are of the form $\{(\partial/\partial h_i)\}$, and the equations of motion are independent of the foot heights h_i.

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

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ l & -l \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{21}$$

[3]Reference [1] shows that the hexapod is small time locally gait controllable when a tripod gait is used.

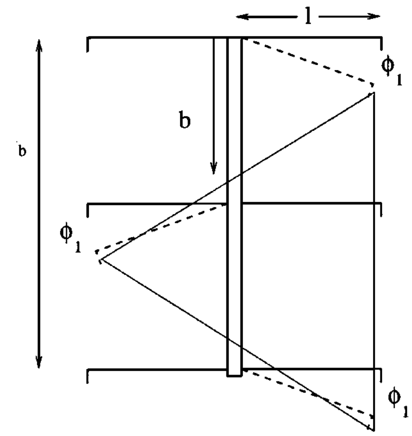

Fig. 5. Stability margin for hexapod tripod gait.

where (x, y, θ) represents the planar position of a reference frame attached to the robot's center, ϕ_1 is the angle of legs 1-4-5 and ϕ_2 is the angle of legs 2-3-6. The variables u_3 and u_4 are both 0 since the legs maintain ground contact. Let $g_{12,1}$ and $g_{12,2}$ represent the first and second columns in (21).

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

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ l & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_4 \end{pmatrix} \tag{22}$$

where h_2 is the height of legs 2-3-6 and u_3 is constrained to be 0. Label columns one, two and three in (22) $g_{1,1}, g_{1,2}$ and $g_{1,3}$, respectively. If legs 2-3-6 are in ground contact and legs 1-4-5 are not, the equations of motion are

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \end{pmatrix} = \begin{pmatrix} 0 & \cos\theta & 0 \\ 0 & \sin\theta & 0 \\ 0 & -l & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{23}$$

where u_4 is constrained to be 0. The columns in (23) are denoted $g_{2,1}, g_{2,2}$, and $g_{2,3}$.

For motion planning purposes, we must select enough vector fields to span the tangent space of the bottom stratum, S_{12}. A simple calculation shows that the set of vector fields, $\{g_{12,1}, g_{12,2}, g_{1,2}, g_{2,1}, [g_{12,1}, g_{12,2}]\}$ spans $T_x S_{12}$ for all $x \in S_{12}$. Note that $[g_{12,1}, g_{12,2}] = (-2l\sin\theta, 2l\cos\theta, 0, 0, 0)^T$.

The *stratified extended system* is constructed from the extended system that uses the vector fields from all strata.

$$\dot{x} = g_{12,1}v_1 + g_{12,2}v_2 + g_{1,2}v_3 + g_{2,1}v_4 + [g_{12,1}, g_{12,2}]v_5 \tag{24}$$

or, in greater detail

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \cos\theta & 0 & 0 & -2l\sin\theta \\ \sin\theta & \sin\theta & 0 & 0 & 2l\cos\theta \\ l & -l & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}.$$

Let the starting and ending configurations be:

$$p = (x, y, \theta, \phi_1, \phi_2, h_1, h_2) = (0, 0, 0, 0, 0, 0, 0)$$
$$q = (x, y, \theta, \phi_1, \phi_2, h_1, h_2) = (1, 1, 0, 0, 0, 0, 0).$$

A path that connects these points is $\gamma(t) = (t, t, 0, 0, 0, 0, 0)$, $t \in [0, 1]$. Equating $\dot{\gamma}(t)$ with with the stratified extended system and solving for the fictitious controls yields

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \frac{1}{2l} \begin{pmatrix} l(\cos\theta + \sin\theta) \\ l(\cos\theta + \sin\theta) \\ -l(\cos\theta + \sin\theta) \\ -l(\cos\theta + \sin\theta) \\ (\cos\theta - \sin\theta) \end{pmatrix}$$

or, since $\theta(t) = 0$, and if we let $l = 1$

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}.$$

For a system which is nilpotent of order 2, we have from (9) (where the g's from (24) are substituted for the b_i's in (9) in the order that they appear in (24)

$$\dot{h}_1 = v_1$$
$$\dot{h}_2 = v_2$$
$$\dot{h}_3 = v_3$$
$$\dot{h}_4 = v_4$$
$$\dot{h}_5 = v_5 + h_1 v_2$$

which yields

$$h_1(1) = h_2(1) = \frac{1}{2}$$
$$h_3(1) = h_4(1) = -\frac{1}{2}$$
$$h_5(1) = \frac{3}{4}.$$

Since the nilpotent approximation is of order two, there is no need to transform to forward Philip Hall coordinates. Instead, we can directly construct a sequence of controls to move in the desired direction.

Let $\circ$ denote concatenation of control inputs, so that, for example, $u_1 \circ u_2$ denotes that $u_1 = 1$ for time $h_1(1)$ followed by $u_2 = 1$ for time $h_2(1)$. Considering the vector fields on S_{12}, ($g_{12,1}, g_{12,2}$ and $[g_{12,1}, g_{12,2}]$), the system needs to flow along the first two vector fields for $(1/2)$ seconds, and construct a piecewise approximation to the flow along the third Lie bracket vector field for $(3/4)$ seconds. The control sequence to approximately move the system in the direction of the flow of the Lie bracket is

$$u_1 \circ u_2 \circ -u_1 \circ -u_2 \tag{25}$$

where each of the individual control inputs is equal to one for $\sqrt{(3/4)}$ seconds [recall (2)]. To flow along $g_{11,1}, u_1 = 1$ for $(1/2)$ seconds. Similarly to flow along $g_{12,1}, u_2 = 1$ for $(1/2)$ seconds.

On the higher strata, to flow along $g_{1,1}, u_1 = -1$ for $(1/2)$ seconds and to flow along $g_{2,1}, u_1 = -1$ for $(1/2)$ seconds.

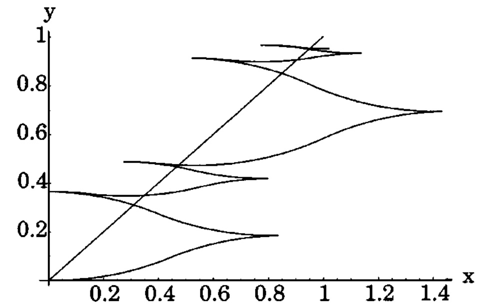

Fig. 6. Straight trajectory.

In order to execute these flows, the robot must switch from the bottom stratum to the higher strata when executing a control input associated with a fictitious input for a higher strata.

Thus, the total control sequence is

$$\sqrt{\frac{3}{4}}(u_1 \circ u_2 \circ -u_1 \circ -u_2) \circ \frac{1}{2}u_2 \circ \frac{1}{2}u_1 \circ \epsilon u_4 \circ \left(-\frac{1}{2}u_2\right) \circ (-\epsilon u_4) \circ \epsilon u_3 \circ -\left(\frac{1}{2}u_1\right) \circ (-\epsilon u_3).$$

The first four terms in the sequence approximate the Lie bracket motion on the bottom stratum. The $\sqrt{(3/4)}$ term denotes the length of time each control input is "on." The next two terms are the contribution of the u_1 and u_2 terms individually on the bottom stratum. The next term represents a small flow associated with removing legs 2-3-6 out of contact with the ground, and the following term corresponds to legs 2-3-6 moving back to their initial position. Since the legs are not in contact with the ground, this motion does not cause the body of the robot to move. The next input corresponds to legs 2-3-6 moving back into contact with the ground. The next three inputs correspond to legs 1-4-5 performing an analogous motion.

Fig. 6 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. Note that the main body axis is oriented along the x-axis in this example. Since the legs can not move immediately sideways, the robot's motion must include "parallel-parking-like" behavior to follow this line.

There is no inherent limitation in the method which requires the trajectory to be broken down into subsegments, however, there are two reasons to do so. First, since the method is based upon decomposing a desired trajectory into flows along the Philip Hall basis vector fields, the final trajectory is only related to the desired trajectory in that the end points are the same (or approximately the same for nilpotent approximations). Breaking the path into segments leads to better overall tracking. Second, robot stability requirements may also demand smaller steps.

The approach is general enough that approximate tracking of arbitrary trajectories is possible. Fig. 7 shows the hexapod following an ellipse while maintaining a constant angular orientation. Fig. 8 shows the results when a smaller step size is used. In the first simulation, the elliptical trajectory is broken into 30

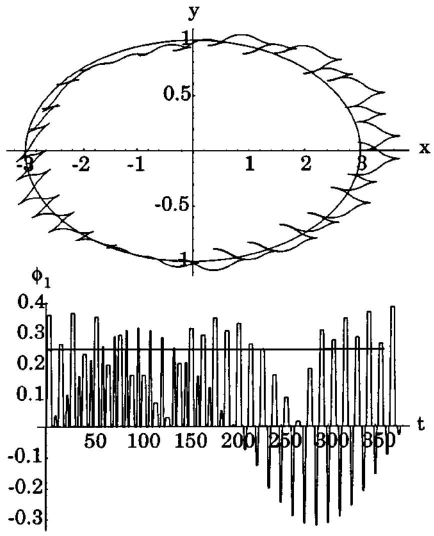

Fig. 7. Elliptical trajectory.

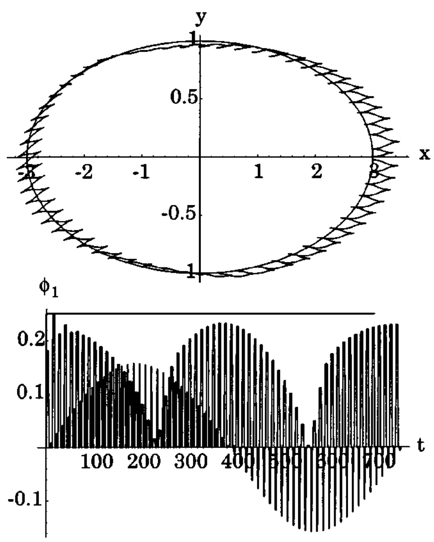

Fig. 8. Elliptical trajectory with smaller steps.

segments. In the second, it is divided into 60 segments. In this example, part of the trajectory tracking error is due to the nilpotent approximation, but another contribution to the error is the simplicity of the model. Some directions are more "difficult" for the system to execute than others due to the kinematic limitations of the leg design. Because this mechanism can not execute "crab-like" gaits, its tracking error during sideways motions increases, as this direction corresponds to a Lie bracket direction.

Also plotted in these figures is the stability criterion. Let the body length be 2 units of length and let the center of mass be located a distance of 0.75 units from the front of the robot. Then, the stability criterion is $\phi_1 < 0.25$ [rad] and $\phi_2 > -.85$ [rad]. In Figs. 7 and 8 the stability limits for ϕ_1 are indicated by the straight horizontal lines. In the first case, where the robot takes bigger steps, the stability condition is violated. However, in the second case it is not.

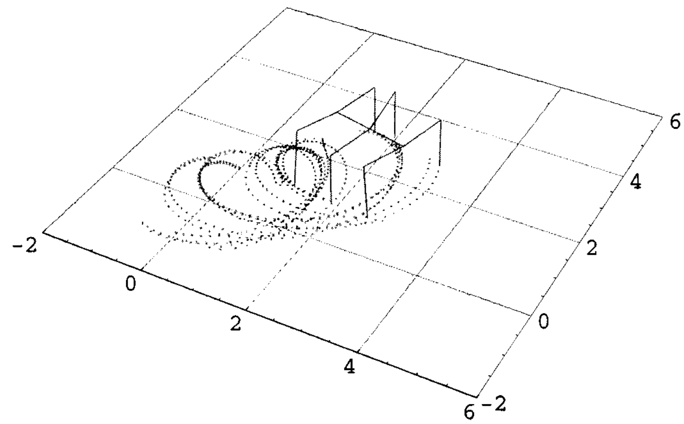

Fig. 9. Hexapod leaving footprints.

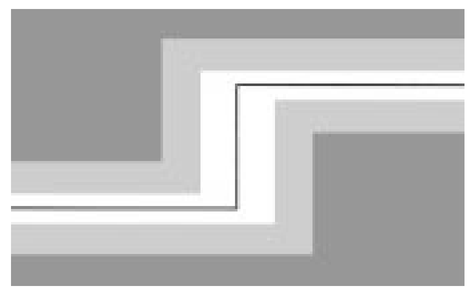

Fig. 10. Nominal obstacle avoidance trajectory.

Fig. 9 depicts the footprints left by the hexapod as it follows a straight line diagonal path while simultaneously rotating at a constant rate. The complex pattern of the footfalls suggests that any technique that is based on foot placement would be very difficult to apply to this system.

Finally, we consider obstacle avoidance. While the nominal initial trajectory $\gamma(t)$ must *a priori* avoid any obstacles, this constraint alone will not guarantee that the actual motion avoids obstacles. If the trajectory is divided into sufficiently small segments, as suggested in Section IV-C, then obstacle avoidance can be realized. Fig. 10 shows a desired nominal path (indicated by a black line) of the hexapod's body center through a set of obstacles. The walls of the environment are indicated by dark grey regions. The lighter grey regions correspond to locations of the robot's center where some vehicle orientations may cause the hexapod to intersect the walls (i.e., the grey regions are the projected silhouettes of the c-space obstacles).

To make the problem more challenging, we also specify that the robot rotates at a uniform rate as it follows the nominal trajectory. A real-world scenario where this might be desirable is a patrol robot that must constantly scan in all directions. Fig. 11 (top) shows the path of robot's center of mass when the trajectory is not finely divided enough to satisfy the criteria of Section IV-C (it is subdivided into 100 subtrajectories). Since the path of the center of mass intersects the lighter grey regions during portions of its motion, the robot would realistically bump into the walls in this example. However, if the nominal trajectory is sufficiently subdivided (into 300 subtrajectories in this case) to

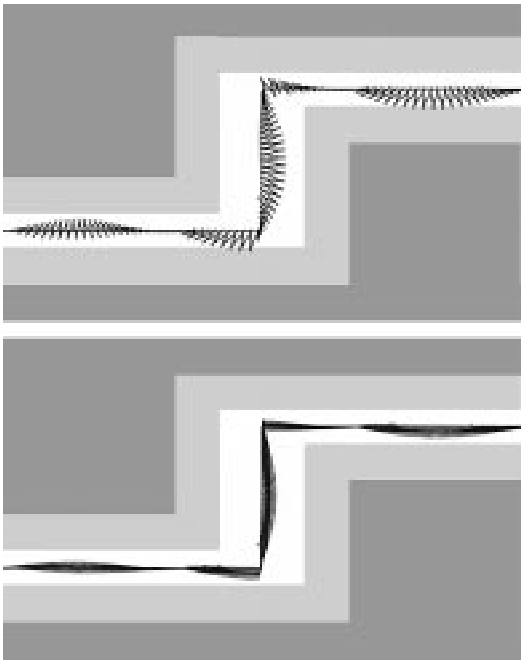

Fig. 11. Top: obstacles not avoided. Bottom: obstacles avoided.

satisfy the requirement of Section IV-C, the robot avoids the walls, as illustrated in the bottom part of Fig. 11.

VI. Multi-Fingered Hand Manipulation

The methodology described above can be almost immediately applied to object manipulation via finger gaiting in a multi-fingered hand as long as the equations of motion can be written in the form of a kinematic system. This may be difficult in the case of rolling contact because the equations of motion may become extremely complicated. Preliminary efforts to overcome this limitation can be found in [36]. The application of this approach leads to an object manipulation planning strategy that is independent of the geometry of the grasped object and independent of the manipulating hand's morphology. The method is also independent of the type of contact between the finger and object (e.g., "point contact with friction," "soft finger" etc.) and independent of the morphology of the manipulating "fingers" (i.e., independent of the number of joints, etc.).

Consider the "egg-shaped" object in Fig. 12 whose surface is parameterized by

$$c(u,v) = \begin{pmatrix} \left(1+\frac{u}{\pi}\right)\cos u \cos v \\ \left(1+\frac{u}{\pi}\right)\cos u \sin v \\ \frac{3}{2}\sin u \end{pmatrix}, \quad \begin{matrix} u \in \left(\frac{-\pi}{2}, \frac{\pi}{2}\right) \\ v \in (-\pi, \pi). \end{matrix}$$

This object is to be manipulated by four, three DOF fingers whose kinematic model is shown in Fig. 13. A "point contact with friction" model is assumed.

The stratified c-space will consist of a total of 16 different strata, corresponding to all the possible combinations of finger contacts. However, as will be clear shortly, the system is manipulable if it is restricted to only five strata: when all four fingers are in contact plus each of the four cases where only one of the fingers is out of contact. Denote these strata as $S_{1234}, S_{123}, S_{124}, S_{134}$, and S_{234} where the subscripts denote which fingers are in contact with the object.

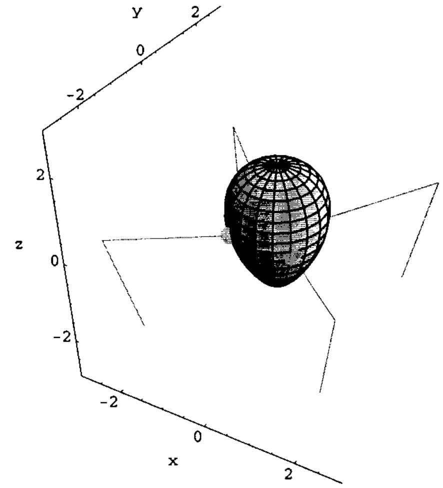

Fig. 12. Four fingers manipulating an object.

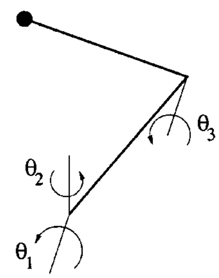

Fig. 13. Finger kinematics.

Since the nominal trajectory stays away from the fingers' kinematic singularities, the finger tip velocities can be considered as system inputs. This input choice will simplify the computations and make the equations of motion satisfy (16). One can not generally choose the inputs in this way, (for example, when the the finger tips are in rolling contact with the object); however, the more general cases still fits within the framework of the stratified motion planning method outlined in Section III.

The equations of motion for such a grasped system are straightforward, though possibly tedious, to derive (see [32] for details). The equations of motion on the bottom stratum are of the form

$$\dot{x} = g_1(x)u_1 + \cdots + g_6(x)u_6$$

and on the higher strata are of the form

$$\dot{x} = g_1(x)u_1 + \cdots + g_6(x)u_6 + g_7(x)u_7 + g_8(x)u_8 + g_9(x)u_9$$

where the first 6 inputs are associated with the finger tip velocities for the three fingers contacting the object, and inputs 7–9 are the three degrees of freedom for the finger that is not in contact with the object. Note that $g_7(x)$ through $g_9(x)$ will take the form $(0, \ldots, 1, \ldots, 0)$ since they are the unconstrained finger tip velocities of the finger which is not contacting the object, and thus they will satisfy (16). Therefore, they may be incorporated into the equations of motion for the bottom stratified extended system.

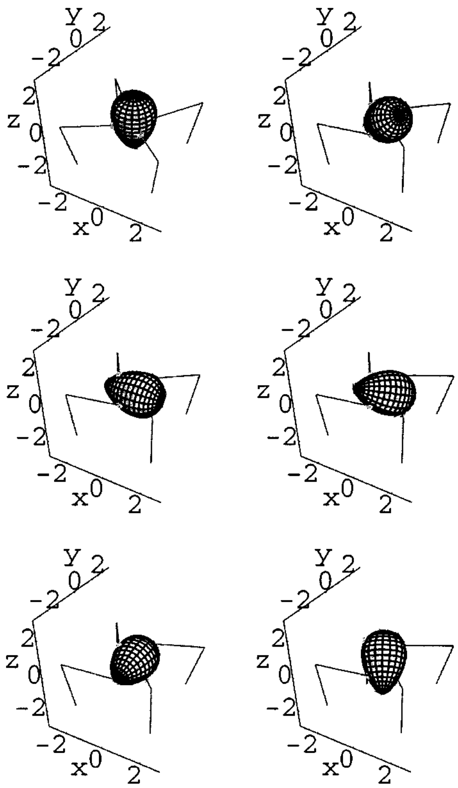

Fig. 14. Snapshots of computer simulation of object manipulation via finger repositioning.

Incorporating these unconstrained finger tip velocity vector fields for each of the four higher strata gives a stratified extended system of the form

$$\dot{x} = \underbrace{g_1(x)u_1 + \cdots + g_6(x)u_6}_{\text{on } S_{1234}} + \underbrace{g_7(x)u_7 + g_8(x)u_8 + g_9(x)u_9}_{\text{from } S_{123}} \\ \vdots \\ + \underbrace{g_{16}(x)u_{16} + g_{17}(x)u_{17} + g_{18}(x)u_{18}}_{\text{from } S_{234}},$$

where all the vector fields except those on the first line correspond to free finger tip motion. Tedious detailed calculations show that $\{g_1, \ldots, g_{18}\}$ spans the tangent space to the c-space, so the system is stratified manipulable. Since no Lie brackets are necessary to make the system stratified manipulable, this system is already in extended form, and the actual control inputs are the same as the "fictitious" inputs presented in Sections II and III.

Assume that the initial and final configurations are identical (as illustrated in Fig. 12), and that the desired motion is a pure rotation of 2π about the axis $\omega = ((1/\sqrt{3}), (1/\sqrt{3}), (1/\sqrt{3}))$. Using exponential coordinates, then, the object's nominal configuration as a function of time is given by Rodrigues' formula:

$$\gamma(t) = e^{\hat{\omega}2\pi t} = I + \hat{\omega}\sin 2\pi t + \hat{\omega}^2(1 - \cos 2\pi t), \qquad t \in [0, 1].$$

For the object's initial and final configuration in Fig. 12, each finger is oriented at an angle of $\pi/4$ relative to the x- and y-axes. As the object rotates, each finger's nominal configuration is such that it contacts the object along that same axis. This can be determined by equating the forward kinematics for each finger with the point on the object's surface that intersects the respective $\pi/4$ radial from the origin, and then, using the kinematics of each finger, determine the desired joint configurations. For this particular example, this trajectory is difficult to compute analytically, but is simple to do numerically for each step of the system's motion. The desired trajectory is decomposed into 10 subsegments, and a sequence of six "snapshots" from the manipulation is shown in Fig. 14.

VII. Conclusion

Our method provides a general means to solve the trajectory generation problem for many types of legged robotic and multi-fingered systems. The simulations indicate that the approach is rather simple to apply. The method is independent of the number of legs (fingers) and is not based on foot (finger) placement principles. For a given legged robot mechanism, a specifically 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 for a quasi-static robot of nearly arbitrary morphology. More importantly, we believe that our approach provides an evolutionary path for future research and generalizations.

Since many interesting 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, it may be difficult to make headway along these lines until more complete results for the smooth case become known.

Acknowledgment

The authors would like to thank some anonymous reviewers for their helpful and insightful comments.

References

[1] B. Goodwine and J. Burdick, "Gait controllability for legged robots," in *Proc IEEE Int. Conf. Robotics and Automation*, 1998.

[2] B. Goodwine and J. W. Burdick, "Controllability of kinematic systems on stratified configuration spaces," *IEEE Trans. Automat. Contr.*, vol. 46, pp. 358–368, 2002.

[3] G. Lafferriere and H. J. Sussmann, "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning*, X. Li and J. F. Canny, Eds. Norwell, MA: Kluwer, 1993, pp. 235–270.

[4] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 700–716, 1993.

[5] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain," in *IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 1991, pp. 1405–1411.

[6] J. K. Lee and S. M. Song, "Path planning and gait of walking machine in an obstacle-strewn environment," *J. Robot. Syst.*, vol. 8, pp. 801–827, 1991.

[7] M. Berkemeire, "Modeling the dynamics of quadrupedal running," *Int. J. Robot. Res.*, vol. 16, no. 9, pp. 971–985, 1998.

[8] S. M. Song and K. J. Waldron, *Machines that Walk: The Adaptive Suspension Vehicle*. Cambridge, MA: MIT Press, 1989.

[9] M. H. Raibert, *Legged Robots that Balance*: MIT Press, 1986.
[10] J. J. Collins and I. Stewart, "Hexapodal gaits and coupled nonlinear oscillator models," *Biolog. Cybernet.*, vol. 68, pp. 287–298, 1993.
[11] C. H. Chen, K. Mirza, and D. E. Orin, "Force control of planar power grasp in the digits system," in *4th Int. Symp.on Robotics and Manufacturing*, 1992, pp. 189–194.
[12] D. Prattichizzo and A. Bicchi, "Dynamic analysis of mobility and graspability of general manipulation systems," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 241–258, Apr. 1998.
[13] E. Rimon and J. W. Burdick, "Configuration space analysis of bodies in contact—i," *Mechanism Machine Theory*, vol. 30, no. 6, pp. 897–912, Aug. 1995.
[14] ——, "Configuration space analysis of bodies in contact—ii," *Mechanism Machine Theory*, vol. 30, no. 6, pp. 913–928, Aug. 1995.
[15] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *Int. J. Robot. Res.*, vol. 15, no. 3, pp. 230–266, 1996.
[16] D. J. Montana, "The kinematics of contact and grasp," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 17–25, 1988.
[17] J. C. Trinkle and R. P. Paul, "Planning for dexterous manipulation with sliding contacts," *Int. J. Robot. Res.*, vol. 9, no. 3, pp. 24–48, 1990.
[18] L. Han, Y. S. Guan, Z. X. Li, Q. Shi, and J. C. Trinkle, "Dextrous manipulation with rolling contacts," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, 1997, pp. 992–997.
[19] R. S. Fearing and T. O. Binford, "Using a cylindrical tactile sensor for determining curvature," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 806–817, Dec. 1991.
[20] K. Salisbury and C. Tarr, "Haptic rendering of surfaces defined by implicit functions," in *Proc. ASME Int. Mechanical Engineering Congress and Exposition*, 1997, pp. 61–67.
[21] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic rendering: Programming touch interation with virtual objects," in *Proc. Symp. on Interactive 3D Graphics*, 1995, pp. 123–130.
[22] T. Okada, "Object handling system for manual industry," *IEEE Trans. Syst., Man Cybernet.*, vol. 9, pp. 79–89, 1979.
[23] M. K. Hor, "Control and task planning of the four finger manipulator," Ph.D dissertation, New York University, New York, 1987.
[24] R. S. Fearing, "Implementing a force strategy for object reorientation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 96–102.
[25] J. Hong and G. Lafferriere, "Fine manipulation with multifinger hands," in *IEEE Int. Conf. on Robotics and Automation*, 1990, pp. 1568–1573.
[26] I.-M. Chen and J. W. Burdick, "A qualitative test for n-finger force-closure grasps on planar objects with applications to manipulation and finger gaits," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 814–820.
[27] B. Goodwine and J. Burdick, "Stratified motion planning with application to robotic finger gaiting," in *Proc. IFAC World Congress*, 1999.
[28] A. Isidori, *Nonlinear Control Systems*, 2nd ed. Berlin, Germany: Springer-Verlag, 1989.
[29] M. Kawski and H. J. Sussmann, "Noncommutative power series and formal lie-algebraic techniques in nonlinear control theory," in *Operators, Systems, and Linear Algebra*, U. Helmke, D. Pratzel-Wolters, and E. Zerz, Eds, Stuttgart: Teubner, 1997, pp. 111–128.
[30] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*. Berlin, Germany: Springer-Verlag, 1984.
[31] J.-P. Serre, *Lie Algebras and Lie Groups*: Springer-Verlag, 1992.
[32] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
[33] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*. Berlin, Germany: Springer-Verlag, 1988.
[34] J. W. Goodwine, "Control of stratified systems with robotic applications," Ph.D. dissertation, California Inst. Technology, Pasadena, 1998.
[35] S. D. Kelly and R. M. Murray, "Geometric phases and robotic locomotion," *J. Robot. Syst.*, vol. 12, no. 6, pp. 417–431, 1995.
[36] B. Goodwine, "Stratified motion planning with application to robotic finger gaiting," in *Proc. 10th World Congress on the Theory of Machines and Mechanisms*, 1999.
[37] G. Lafferriere and H. J. Sussmann, "Motion planning for controllable systems without drift," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, 1991.
[38] H. J. Sussmann, "A product expansion for the Chen series," in *Theory and Applications of Nonlinear Control Systems*, C. I. Byrnes and A. Lindquist, Eds. Amsterdam, The Netherlands: Elsevier, 1986, pp. 323–335.
[39] R. M. Murray, "Nilpotent bases for a class of nonintegrable distributions with applications to trajectory generation for nonholonomic systems," *Math. Control Signals Syst.*, vol. 7, pp. 58–75, 1994.

Bill Goodwine received the B.S. degree in mechanical engineering from the University of Notre Dame, Notre Dame, IN, the J.D. degree from Harvard University, Cambridge, MA, and the M.S. and Ph.D. degrees in applied mechanics from the California Institute of Technology, Pasadena.

His current research interests are in hybrid and stratified control systems, robotic locomotion, and distributed control.

Dr. Goodwine has been the recipient of the National Science Foundation CAREER award.

Joel W. Burdick received the B.S. degrees in mechanical engineering and chemistry from Duke University, Chapel Hill, NC, and the M.S. and Ph.D. degrees in mechanical engineering from Standford University, Stanford, CA.

He joined the department of Mechanical Engineering at the California Institute of Technology, Pasadena, in May 1988 and is currently a professor of Mechanical Engineering and Bioengineering. His current research interests include robotic locomotion, sensor based robot motion planning, multi-fingered robotic grasping, neural prosthetics, and applied nonlinear control theory.

Prof. Burdick has been the recipient of the National Science Foundation Presidential Young Investigator award, the Ofiice of Naval Research Young Investigator award, and the Feynman fellowship. He has also received the ASCIT award for excellence in undergraduate teaching and the GSA award for excellence in graduate student education.