A Modal Approach to Hyper-Redundant Manipulator Kinematics

Gregory S. Chirikjian and Joel W. Burdick

Abstract— **This paper presents novel and efficient kinematic modeling techniques for "hyper-redundant" robots. This approach is based on a "backbone curve" that captures the robot's macroscopic geometric features. The inverse kinematic, or "hyper-redundancy resolution," problem reduces to determining the time varying backbone curve behavior. To efficiently solve the inverse kinematics problem, we introduce a "modal" approach, in which a set of intrinsic backbone curve shape functions are restricted to a modal form. The singularities of the modal approach, modal non-degeneracy conditions, and modal switching are considered. For discretely segmented morphologies, we introduce "fitting" algorithms that determine the actuator displacements that cause the discrete manipulator to adhere to the backbone curve. These techniques are demonstrated with planar and spatial mechanism examples. They have also been implemented on a 30 degree-of-freedom robot prototype.**

I. Introduction

WE USE the word *hyper-redundant* to denote robot manipulators that have a large or infinite degree of kinematic redundancy. These robots are analogous in shape and operation to snakes, elephants' trunks, or tentacles. Their highly articulated structures make them well suited for niche applications, such as inspection and operation in highly constrained environments. Hyper-redundant robots can also be used to implement novel means of worm- or snake-like locomotion and tentacle-like grasping [3]. Fig. 1 shows a 30 degree-of-freedom (DOF) planar robot constructed by the authors [5], [7].

Hyper-redundant robots have previously been called "swan's neck" [8], "tentacle" [9], [10], "highly redundant" [11], "tensor-arm" [17], "elephant trunk" [12], "active cord" [13], [14], "snake-like" [15], and "spine" [16]. The term "hyper-redundant," coined by the authors in [2], is used to describe this class of robotic systems. To our knowledge, the earliest hyper-redundant robot designs and implementations date to the late 1960's [17]. Hirose and coworkers (see [13], [14], and references therein) have implemented a large number of working hyper-redundant systems.

Manuscript received March 24, 1992; revised March 26, 1993. This work was sponsored by National Science Foundation grant #MSS-901779; NSF Presidential Young Investigator award #MSS-9157843; by the Office of Naval Research Young Investigator Award N00014-92-J1920; and by a NASA Graduate Student Researchers Program fellowship (for the first author).

Gregory S. Chirikjian is with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD.

Joel W. Burdick is with the School of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125.

IEEE Log Number 9215094.

Fig. 1. 30 degree-of-freedom hyper-redundant robot.

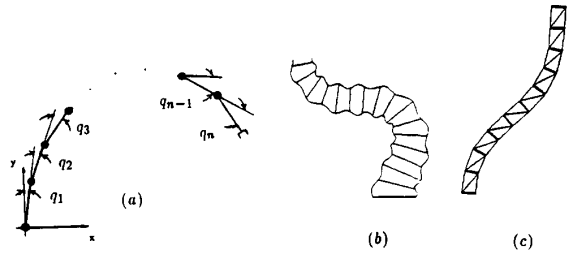


Fig. 2. Different hyper-redundant robot morphologies.

Hyper-redundant manipulators can be implemented in a variety of physical morphologies, including pneumatic bellows [18], a serial chain consisting of a large number of rigid links, a variable geometry truss [19], etc. [21], [22]. The possible morphologies can be roughly categorized into three main types (Fig. 2). In Fig. 2(a), a discrete morphology manipulator is shown, i.e., one with a large, but finite, number of rigid links. Fig. 2(b) shows a continuous morphology; i.e., one whose actuation is distributed over the robot's length and whose shape is continuously deformable. Fig. 2(c) shows a variable geometry truss structure, or VGT, which is a cascade of parallel platforms. VGT's are also a discrete morphology, but are representative of "modular" design approaches. These systems have some decidedly different characteristics from the morphologies of Fig. 2(a) (see Section VI). The robot in Fig. 1 is a VGT design.

This paper presents new and efficient kinematic methods that are suitable for nearly all hyper-redundant robot morphologies. These methods are based on a backbone curve that captures the hyper-redundant robot's important macroscopic features. The inverse kinematic problem (or "hyper-redundancy resolution") reduces to determining the proper time-varying backbone curve behavior. We introduce a hyper-redundancy resolution scheme that is based on the restriction of a set of intrinsic backbone curve shape functions to a "modal" form. This is an arbitrary, though quite useful, restriction, as it leads to efficient inverse kinematic solutions. Further, many hyper-redundant tasks, such as locomotion and grasped object reorientation, are well suited to the modal approach [2], [3]. This paper focuses on the basic backbone curve kinematic modeling method and the details of the modal hyper-redundancy resolution technique. In [4] we determine backbone curve shape based on a user-defined optimality criterion.

"Fitting" algorithms are developed that allow the basic backbone curve analysis to be applied to different physical implementation morphologies. These techniques lead to very efficient inverse kinematic algorithms and form the foundation for efficient and novel obstacle avoidance [2], locomotion [3], and grasping schemes [3]. These methods have been implemented on the robot in Fig. 1, and are practically quite effective [5], [7].

Section III establishes the backbone curve model and introduces related backbone curve parametrizations. Section IV introduces the modal hyper-redundancy resolution scheme. Section V analyses the algorithmic singularities associated with the modal approach, "degenerate" modes, and mode switching algorithms. Section VI describes the "fitting algorithms" needed to apply this method to discrete morphologies.

II. Relation to Previous Work

Most redundancy resolution schemes are based on the manipulator Jacobian pseudo-inverse [23], an extended Jacobian inverse [27], or an augmented inverse Jacobian [28]. However, these techniques are not suited to hyper-redundant robots for a number of practical reasons. The computational burden of the Jacobian pseudo-inverse becomes prohibitive for hyper-redundant robots, which may have hundreds of degrees of freedom. The augmented and extended Jacobian techniques are additionally impractical because an unreasonably large number of additional task constraints must be specified. Further, these Jacobian based techniques are only suitable for serial link morphologies, and not for continuous or VGT morphologies [Fig. 2(b) and (c)]. Finally, these approaches are not a suitable framework for implementing locomotion, tentacle-like grasping, or other tasks that are not focused on the motion of the end-effector. The hyper-redundancy resolution algorithms developed in this paper *are not* based on a Jacobian matrix. Further, the framework is not only useful for motion planning, but also for analyzing and implementing locomotion and other task behaviors that are unique to hyper-redundant robots [2], [3].

Other investigators have developed algorithms for VGT manipulators [11], [19]. In these works, a continuous curve model was used to describe the macroscopic truss geometry. However, a curve alone is not sufficient to uniquely describe a spatial manipulator configuration. We introduce a roll distribution to overcome this problem. Second, unless the curve used to describe the manipulator is parameterized with physically meaningful variables, additional complex computations are required to specify actuator displacements and ensure that joint or stroke limits are not exceeded. Last, while [11], [19] deal exclusively with VGT manipulators, it is not clear how they would apply to other types of hyper-redundant morphologies. Other authors have proposed related ideas for strictly planar inextensible manipulators [9], [20]. Our method is general for both planar and spatial mechanisms, and can handle *extensible* manipulators, a novel feature not previously considered in the literature.

III. Hyper-Redundant Robot Kinematics: The Backbone Curve Approach

We consider only hyper-redundant robot structures without macroscopic branches or closed loops. Small internal closed loops, such as those of the VGT in Fig. 2(c), can be accounted for in this method.

Definition: A *backbone curve* is a piecewise continuous curve that captures the important macroscopic geometric features of a hyper-redundant robot.

The backbone curve directly captures the geometry of continuous morphology robots, and is typically the robot centerline or spine. For discretely segmented robots, we assume that the robot is comprised of a sufficiently large number of segments, modules, or links so that the backbone curve nominally captures the robot geometry closely, though not exactly. A backbone curve is not sufficient to represent the microscopic properties of spatial hyper-redundant manipulators. A complete representation also requires a reference frame at each backbone curve point.

Definition: A *backbone curve reference set* consists of a backbone curve parametrization and an associated set of orthonormal frames.

In this modeling paradigm, task planning reduces to determining the time varying backbone curve reference set geometry that satisfies task requirements, such as end-effector positioning, obstacle avoidance, or locomotion. The resulting backbone curve shape can be used directly to control the geometry of a continuous morphology robot. For discretely segmented manipulators, we apply "fitting algorithms" (Section VI).

A. Parametrizations of the Backbone Curve

We seek to parametrize the Cartesian location of backbone curve points in the form $\overline{x}(s,t)$, where s is a parameter (not necessarily the classical arc-length parameter) that measures the distance along the backbone curve at time t. We further wish to parametrize the backbone curve in ways that can incorporate important physical properties and limitations of the robot structure. The classical geometry and computer graphics

literature provide several possible intrinsic and extrinsic backbone curve parameterization schemes [24], [26]. We focus on intrinsic schemes, as they define shape with a minimal set of parameters and are invariant under change of reference frame. Extrinsic schemes, such as splines, can not easily incorporate important physical constraints, such as fixed robot length or bounded curvature.

1) Classical Geometry of Curves: In the classical Frenet-Serret Apparatus [24], inextensible spatial curves are modeled by a vector function $\hat{\overline{x}}(L) \in \mathbb{R}^3$. L is an arclength measure of distance along the backbone curve: $|d\hat{\overline{x}}/dL| = 1$. For every curve satisfying $|d^2\hat{\overline{x}}/dL^2| \neq 0$, a unique Frenet-Serret frame can be assigned at each point L. The unit basis vectors of this frame are:

$$\overline{u}(L) = \frac{d\hat{\overline{x}}}{dL}; \quad \overline{n}(L) = \frac{1}{\kappa(L)}\frac{d\overline{u}(L)}{dL}; \quad \overline{b}(L) = \overline{u}(L) \times \overline{n}(L) \tag{1}$$

where $\overline{u}(L)$, $\overline{n}(L)$, and $\overline{b}(L)$ are respectively termed the tangent, normal, and binormal vectors. $\kappa(L)$ is the curvature function of the backbone curve:

$$\kappa^2(L) = \frac{d^2\hat{\overline{x}}(L)}{dL^2} \cdot \frac{d^2\hat{\overline{x}}(L)}{dL^2}. \tag{2}$$

It can be shown [24] that the curvature function and the torsion function:

$$\tau(L) = \frac{1}{\kappa^2(L)}\overline{u}(L) \cdot \left(\frac{d\overline{u}(L)}{dL} \times \frac{d^2\overline{u}(L)}{dL^2}\right) \tag{3}$$

uniquely determine, up to rigid body displacement, the geometry of a fixed-length curve.

While the Frenet-Serret system is physically intuitive and sometimes required for hyper-redundant robot analysis, is often not useful for practical applications. First, the Frenet-Serret frame is not defined when $\kappa = 0$ (e.g., straight line segments). Second, the computation of $\hat{\overline{x}}(L)$, given $\kappa(L)$ and $\tau(L)$, requires the numerical solution of the following differential equation:

$$\overline{u}' - \left(2\frac{\kappa'}{\kappa} + \frac{\tau'}{\tau}\right)\overline{u} + \left(\kappa^2 + \tau^2 - \frac{\kappa\kappa - 2(\kappa')^2}{\kappa^2} + \frac{\kappa'\tau'}{\kappa\tau}\right)\overline{u}' + \kappa^2\left(\frac{\kappa'}{\kappa} - \frac{\tau'}{\tau}\right)\overline{u} = 0 \tag{4}$$

where $'$ indicates differentiation with respect to L. To overcome these limitations, we introduce a new parametrization that is useful for hyper-redundant backbone curves.

B. An Integral Representation

Points of extensible spatial curves can also be parametrized as follows:

$$\overline{x}(s,t) = \int_0^s l(\sigma,t)\overline{u}(\sigma,t)d\sigma \tag{5}$$

$\overline{u}(s,t)$ is the unit vector tangent to the curve at s. Unless otherwise specified, in this paper we use the convention: $\overline{u}(0,t) = [0,1,0]^T$. $l(s,t)$ is a scaling factor that controls the length of the curve tangent and assumes the general form:

$$l(s,t) = 1 + \epsilon(s,t) > 0 \tag{6}$$

$\epsilon(s,t)$ is the local extensibility of the manipulator at point s and time t. $\epsilon(s,t) > 0$ indicates a local extension at s, while $\epsilon(s,t) < 0$ corresponds to local contraction. The arc-length of the backbone curve between points s_1 and s_2 is:

$$L(s_2,t) - L(s_1,t) = \int_{s_1}^{s_2} l(\sigma,t)d\sigma. \tag{7}$$

The parametrization of (5) has the following interpretation. The backbone curve is "grown" from the base by propagating the curve forward along the tangent vector, which varies its direction according to $\overline{u}(s,t)$ and varies its magnitude (or "growth rate") according to $l(s,t)$. Any parametrization of spherical kinematics (such as Euler angles and quaternions) can be used to parameterize $\overline{u}(s,t)$, and thus form the basis for a backbone curve representation scheme. Hereafter we use the following backbone curve representation:

$$\overline{x}(s,t) = [x_1(s,t)\ x_2(s,t)\ x_3(s,t)]^T = \begin{pmatrix} \int_0^s l(\sigma,t)\sin K(\sigma,t)\cos T(\sigma,t)d\sigma \\ \int_0^s l(\sigma,t)\cos K(\sigma,t)\cos T(\sigma,t)d\sigma \\ \int_0^s l(\sigma,t)\sin T(\sigma,t)d\sigma \end{pmatrix}. \tag{8}$$

The definitions of $K(s,t)$ and $T(s,t)$ are shown in Fig. 3. By convention, $K(0,t) = T(0,t) = 0$ is assumed.

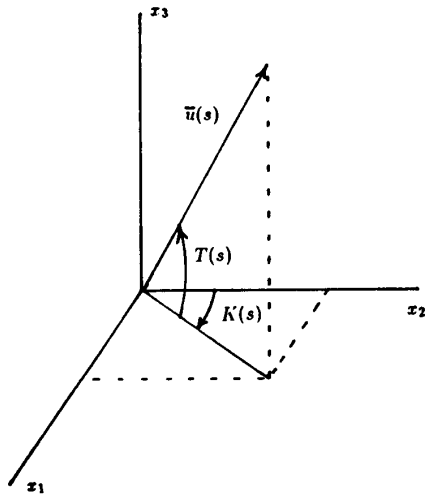


Fig. 3. Definition of $K(s,t), T(s,t)$.

The kinematics of planar curves is a degenerate case of (8) with $T(s,t) = 0 \forall s$. To distinguish the planar case, we use the symbol $\theta(s,t)$ instead of $K(s,t)$, where $\theta(s,t)$ is the clockwise measured angle that the tangent to the planar curve makes with the x_2-axis at time t. Note that $\kappa(s,t) = (1/l(s,t))\partial\theta(s,t)/\partial s$ in the planar case, and $\overline{u}(s,t) = [\sin\theta(s,t), \cos\theta(s,t)]^T$.

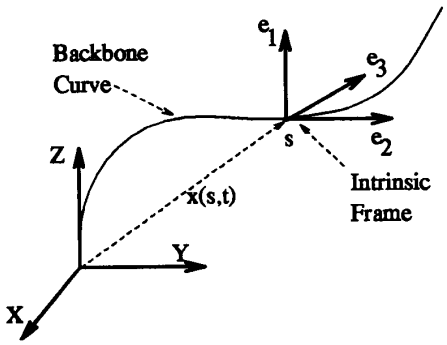


Fig. 4. Backbone reference frame.

The Frenet-Serret parametrization and this parametrization can be related as follows (where $(\cdot)$ represents a derivative with respect to s):

$$\kappa^2 = \frac{1}{l^2}[(\dot{T})^2 + (\dot{K})^2 \cos^2 T]$$
$$\tau = \frac{1}{l}\left[-\dot{K} \sin T + \frac{1}{\kappa^2}[\dot{T}\ddot{K} - \ddot{T}\dot{K}) \cos T - (\dot{T})^2 \dot{K} \sin T]\right] \tag{9}$$

C. Parametrization of the Backbone Curve Reference Frame

A backbone reference frame at s has right-handed orthonormal basis vectors, $\{\overline{e}_1, \overline{e}_2, \overline{e}_3\}$, and its origin coincides with point $\overline{x}(s,t)$. The set of backbone frames can be parametrized as:

$$\mathbf{Q}(s,t) = (\overline{e}_1(s,t)\overline{e}_2(s,t)\overline{e}_3(s,t)) \in SO(3) \tag{10}$$

where $\mathbf{Q}(0,t) = \mathbf{I}$. There is much freedom in the assignment of backbone reference frames. A backbone curve parametrization will typically have a set of frames naturally associated with it, as in the Frenet-Serret case. We call this frame the *parametrization induced reference frame,* or *induced frame,* and denote it by $\mathbf{Q}_{IR}(s,t)$. For the parametrization of Fig. 3, we assign to every s the frame with whose orientation is described by (11) at the foot of this page, where $\mathbf{Q}_{IR}(0,t) = I$. The induced frame should not be confused with the backbone reference frame. It can differ from the backbone reference frame by an s-dependent twist about the backbone curve tangent, which arises as follows.

The macroscopic geometry of spatial backbone curves can not be completely specified by the backbone curve alone. For example, Fig. 5 shows two spatial VGT's having the same backbone curves, but different twists about the backbone. Thus, a roll distribution, $R(s,t)$, is also required to completely specify a spatial hyper-redundant robot configuration. $R(s,t)$ measures how $\mathbf{Q}(s,t)$ twists about the backbone curve with respect to $\mathbf{Q}_{IR}(s,t)$ is formally defined

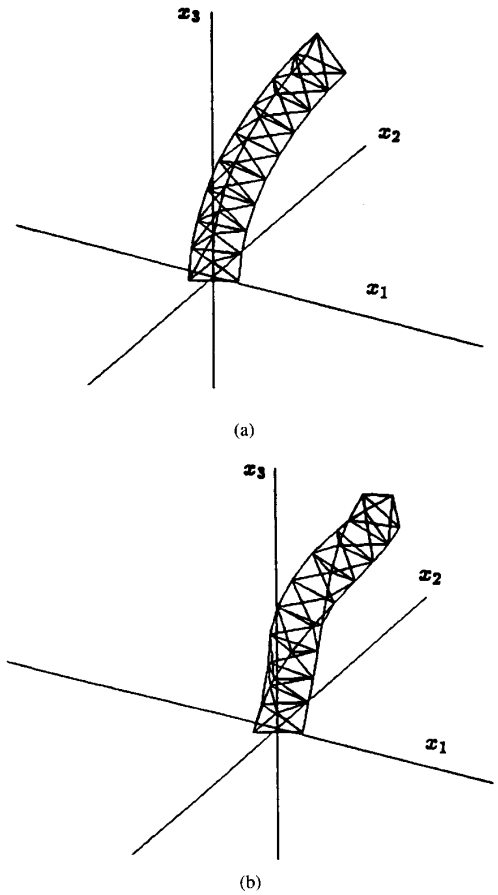


Fig. 5. Two different VGT's with the same backbone curve.

$$\mathbf{Q}(s,t) = \mathbf{Q}_{IR}(s,t)\mathrm{Rot}(\overline{u}(s,t), R(s,t)) \tag{12}$$

where $\mathrm{Rot}(\overline{v}, \phi)$ is rotation about axis $\overline{v}$ by angle ϕ. $R(0,t) = 0$ is assumed.

In summary, the backbone curve reference set, which consists of the backbone curve and associated set of orthonormal frames, is a function of a small set of shape functions. For example, a spatial backbone curve reference set can be completely described by four independent functions $l(s,t), K(s,t), T(s,t)$, and $R(s,t)$. The choice of shape function basis is not unique, as $l(s,t), \kappa(s,t), \tau(s,t)$, and $R(s,t)$ can also parametrize a spatial backbone curve. Some parametrizations are most useful for insight, while others are most useful for efficient computation.

$$\mathbf{Q}_{IR}(s,t) = \begin{pmatrix} \cos K(s,t) & \sin K(s,t) \cos T(s,t) & -\sin K(s,t) \sin T(s,t) \\ -\sin K(s,t) & \cos K(s,t) \cos T(s,t) & -\cos K(s,t) \sin T(s,t) \\ 0 & \sin T(s,t) & \cos T(s,t) \end{pmatrix} \tag{11}$$

IV. A Modal Approach to Hyper-Redundancy Resolution

We define the inverse kinematic problem (or "hyper-redundancy resolution") as the problem of finding a backbone curve reference set (or its equivalent shape functions) that satisfies task constraints. This paper mainly considers the constraint of end-effector positioning. There are generically an infinite number of solutions. Optimality criteria could be used to choose the curve that satisfies task constraints while optimizing additional criterion. We have pursued this approach in [4]. This section introduces a modal approach for computing hyper-redundant inverse kinematic solutions. This scheme is computationally efficient and well suited to many hyper-redundant robot robot tasks.

A. The Modal Shape Function Method

Let $\{\mathcal{S}_i(s,t)\}$ for $i \in \{1,2,3,4\}$ denote the set of backbone curve reference set shape functions. For example, $\mathcal{S}_1(s,t) = l(s,t), \mathcal{S}_2(s,t) = K(s,t), \mathcal{S}_3(s,t) = T(s,t), \mathcal{S}_4(s,t) = R(s,t)$. In the modal approach, we restrict each $\mathcal{S}_i(s,t)$ to the form:

$$\mathcal{S}_i(s,t) = \sum_{j=1}^{N_{\mathcal{S}_i}} a_{ij}(t)\phi_{ij}(s) \tag{13}$$

where the $\{\phi_{ij}(s)\}$ are mode functions, while the $\{a_{ij}(t)\}$ are modal participation factors. $N_{\mathcal{S}_i}$ is the number of modes distributed in the ith shape functions, while $\Sigma_i N_{\mathcal{S}_i}$ is the total number of modes. This number must equal or exceed the number of geometric constraints dictated by the task. In general, the number of modes is far less than the number of mechanical degrees of freedom in the mechanism. Hereafter we denote the set of modal participation factors by the vector $\bar{a}$. The $\{\phi_{ij}\}$ are predetermined functions chosen by the robot programmer (see Section V for restrictions in choosing the $\{\phi_{ij}\}$). Consequently, for a given set of modes, the backbone curve shape is determined solely by the $\{a_{ij}\}$. Thus, the inverse kinematic problem reduces to the search for the proper $\{a_{ij}\}$ that satisfy the task constraints. Several examples will illustrate the method and its generality.

First consider an inextensible planar backbone curve, which is uniquely defined by the single shape function $\theta(s)$ [or equivalently, $\kappa(s)$]. $\theta(s,t)$ is restricted to the form: $\theta(s,t) = \Sigma_{i=1}^{N_\theta} a_i(t)\phi_i(s)$. For some mode choices, closed form inverse kinematic solutions can be found. For example, consider the following choice of modes for $N_\theta = 2$ (a planar task where only the end-effector position is of concern):

$$\phi_1(s) = \sin 2\pi s; \qquad \phi_2(s) = 1 - \cos 2\pi s. \tag{14}$$

Substituting these two modes into (13) and then into (8) and evaluating at $s = 1$, it can be shown, using identities in [25], that the "forward" kinematic equations reduce to:

$$\begin{aligned} x_{ee}(t) &= x_1(1,t) = x_1 \text{ tip position} \\ &= \sin(a_2(t))J_0[(a_1^2(t) + a_2^2(t))^{1/2}] \\ y_{ee}(t) &= x_2(1,t) = x_2 \text{ tip position} \\ &= \cos(a_2(t))J_0[(a_1^2(t) + a_2^2(t))^{1/2}] \end{aligned} \tag{15}$$

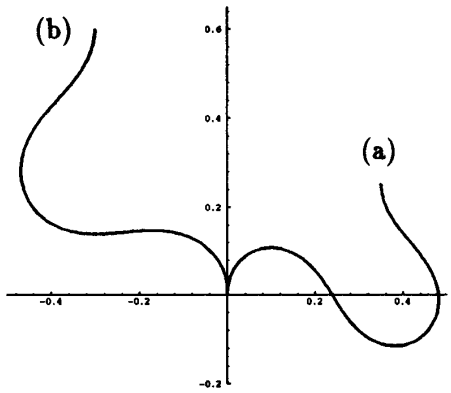


Fig. 6. Planar example.

where $J_0(\cdot)$ is the 0th order Bessel function. (15) can be symbolically inverted to solve for the "inverse kinematic" solution (the modal participation factors for given (x_{ee}, y_{ee}):

$$\begin{aligned} a_1(t) &= \pm([J_0^{-1}[(x_{ee}^2(t) + y_{ee}^2(t))^{1/2}]]^2 \\ &\qquad - [\text{Atan2}(x_{ee}(t), y_{ee}(t))]^2)^{1/2} \\ a_2(t) &= \text{Atan2}(x_{ee}(t), y_{ee}(t)). \end{aligned} \tag{16}$$

$J_0^{-1}(\cdot)$ is the "restricted inverse Bessel function of zero order, v" and is defined as the inverse of $J_0(z)$ for $0 < z < \mu$ where $\mu \approx 3.832$ is the first local minimum of J_0. The plus and minus sign distinguishes two possible "poses." Other poses would exist if other intervals of the argument were permitted in computing the inverse of $J_0(z)$.

Fig. 6(a) illustrates this solution when the desired end-effector location is $(x_{ee}, y_{ee}) = (0.35, 0.24)$ and the "+ " pose is chosen. The corresponding modal participation factors are $a_1 = 1.3416$ and $a_2 = 0.9505$. Fig. 6(b) shows another example with $(x_{ee}, y_{ee}) = (-0.3, 0.6)$ and the choice of the "$-$" pose: $a_1 = -1.1075, a_2 = -0.4636$. Other closed form solutions for planar inextensible backbone curves can be found in [6].

When the planar backbone curve is extensible, it can be parametrized by the two shape functions $l(s,t)$ and $\theta(s,t)$, which we restrict to the modal form:

$$\begin{aligned} \theta(s,t) &= \sum_{i=1}^{N_\theta} a_i(t)\phi_i(s) \\ l(s,t) &= \sum_{i=1}^{N_l} a_{\{i+n_\theta\}}(t)\phi_{\{i+N_\theta\}}(s). \end{aligned} \tag{17}$$

An entire class of closed form forward and inverse kinematic solutions that employs both extension and bending modes can be defined as follows. Consider, for $N_\theta = N_l = 1$, the bending and extension modes:

$$\phi_1(s) = v(s) \qquad \phi_2(s) = v'(s) \tag{18}$$

where $v(s)$ is a strictly increasing function ($v'(s) > 0$ for all $s \in [0,1]$) with $v(0) = 0$ and $v(1) = 1$. The forward

kinematics for this class of modes is:

$$\begin{aligned} x_{ee}(t) &= \int_0^1 a_2(t)\phi_2(s)\sin(a_1(t)\phi_1(s))ds \\ &= \frac{a_2(t)}{a_1(t)}(1-\cos a_1(t)) \\ y_{ee}(t) &= \int_0^1 a_2(t)\phi_2(s)\cos(a_2(t)\phi_1(s))ds \\ &= \frac{a_2(t)}{a_1(t)}\sin a_1(t). \end{aligned} \tag{19}$$

The inverse kinematic solution is:

$$\begin{aligned} a_1(t) &= 2\,\mathrm{Atan2}(x_{ee}(t), y_{ee}(t)), \\ a_2(t) &= \frac{a_2(t)y_{ee}(t)}{\sin a_1(t)}. \end{aligned} \tag{20}$$

Examples of this class of modes are shown in Fig. 7, where $(x_{ee}, y_{ee}) = (0.5, 0.5)$. In Fig. 7(a), $v(s) = s$, and in Fig. 7(b) $v(s) = 2/3(1/2s^2+s)$. For both choices of $v(s)$, the backbone curve will be a circular arc, with modal participation factors $a_1 = 1.5708, a_2 = 0.7854$. However, the local extensibility differs in these two cases, so that the distribution of actuator extensions in a real robot will depend on the choice of $v(s)$. The hash marks in Fig. 7 are spaced at equal intervals of s, illustrating the differences in local extensibility.

The modal method can be similarly used to formulate spatial hyper-redundant manipulator kinematic algorithms. Consider positioning the tip of a nonextensible backbone curve in $\mathbb{R}^3$. Let the $K(s,t), T(s,t)$ shape functions assume the form:

$$\begin{aligned} K(s,t) &= a_1(t)\sin(2\pi s) - a_2(t)\cos(2\pi s); \\ T(s,t) &= a_3(t)\sin(2\pi s) - a_4(t)\cos(2\pi s). \end{aligned} \tag{21}$$

The forward kinematic equations corresponding to these modes are (dropping the dependence on t):

$$\begin{aligned} x_{ee} &= \frac{J_0}{2}[\sqrt{(a_1+a_3)^2+(a_2+a_4)^2}]\sin(a_2+a_4) \\ &\quad + \frac{J_0}{2}[\sqrt{(a_1-a_3)^2+(a_2-a_4)^2}]\sin(a_2-a_4) \\ y_{ee} &= \frac{J_0}{2}[\sqrt{(a_1+a_3)^2+(a_2+a_4)^2}]\cos(a_2+a_4) \\ &\quad + \frac{J_0}{2}[\sqrt{(a_1-a_3)^2+(a_2-a_4)^2}]\cos(a_2-a_4) \\ z_{ee} &= J_0[\sqrt{a_3^2+a_4^2}]\sin a_4 \end{aligned} \tag{22}$$

There is no closed-form inverse kinematic solution for these modes. However, Section IV-C introduces a numerical technique for solving this problem. Fig. 8(a) shows a solution obtained using this method for the case $(x_{ee}, y_{ee}, z_{ee}) = (0.5, 0.3, 0.5)$. The modal participation factors are $a_1 = 0.5812, a_2 = 0.83501, a_3 = 0.3718, a_4 = 0.6063$. Note that four modes are defined for a task with only three constraints. The extra mode can be used to modify the internal geometry of the backbone curve for fixed end-effector location. This phenomena might be called a "modal self-motion," Fig. 8(b), (c), and (d) illustrates some of the different possible solutions.

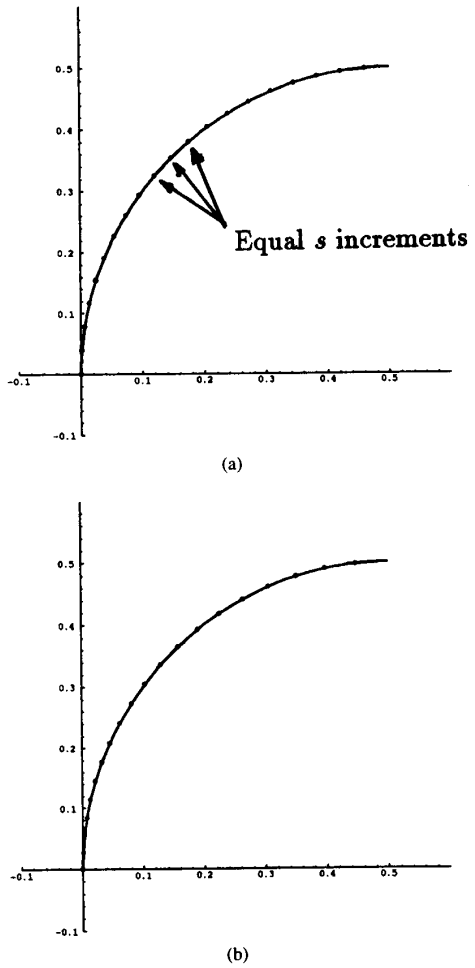


Fig. 7. An example of Planar bending and extension modes.

The kinematic control of orientation can be easily handled in this framework. Note that $\mathbf{Q}(1,t)$ is the orientation of the backbone curve tip. Let $\mathbf{Q}_{ee}(t)$ denote the desired tip orientation matrix, with elements $\{q_{ee,ij}\}$. If $\mathbf{Q}(s,t)$ is parametrized using (12) and (11), then it can be shown that:

$$\begin{aligned} T(1,t) &= \sin^{-1}(q_{ee,32}(t)); \\ K(1,t) &= \mathrm{atan2}\left(\frac{q_{ee,12}(t)}{\cos T(1,t)}, \frac{q_{ee,22}(t)}{\cos T(1,t)}\right) \\ R(1,t) &= \mathrm{atan2}\left(-\frac{q_{ee,33}(t)}{\cos T(1,t)}, \frac{q_{ee,33}(t)}{\cos T(1,t)}\right). \end{aligned} \tag{23}$$

Thus, control of the tip orientation only requires that $K(1,t), T(1,t)$, and $R(1,t)$ assume specified values.

B. Non-Smooth and Discontinuous Modes

Note that the shape functions need not be smooth. For example, consider again a planar nonextensible backbone curve with:

$$\theta(s,t) = a_1(t)H(s-L_1) + a_2(t)H(s-L_2) + a_3(t)H(s-L_3) \tag{24}$$

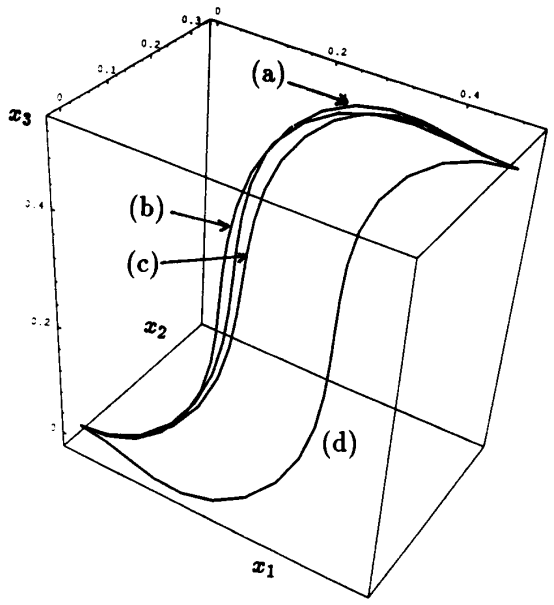


Fig. 8. Example of a spatial modal solution.

where $0 < L_1 < L_2 < L_3 < 1$ are constants and $H(\cdot)$ is the Heaviside unit step function. The equivalent curvature description is $\kappa(s,t) = a_1\delta(s-L_1)+a_2\delta(s-L_1)+a_3\delta(s-L_1)$; i.e., three delta functions located at $s = L_1, s = L_2$, and $s = L_3$. Substituting these modes into the forward kinematics equations, one finds that:

$$x_{ee} = (L_2 - L_1)\sin a_1 + (L_3 - L_2)\sin(a_1 + a_2) + (1 - L_3)\sin(a_1 + a_2 + a_3)$$
$$y_{ee} = L_1 + (L_2 - L_1)\cos a_1 + (L_3 - L_2)\cos(a_1 + a_2) + (1 - L_3)\cos(a_1 + a_2 + a_3)$$

This is exactly the forward kinematics of a 3 link rigid link robot with link lengths $L_2 - L_1, L_3 - L_2$, and $1 - L_3$. In other words, by proper choice of non-smooth and discontinuous shape functions, one can "mimic" traditional rigid link robot kinematics using these continuous curve models. See [6] for additional examples of this phenomena, including examples of how this method can represent prismatic joints.

C. Numerical Solutions to the Modal Inverse Kinematics Problem

This method does not require closed-form forward and inverse kinematic solutions. Many nonlinear numerical equation-solving techniques can be employed to find inverse modal solutions. Practically, lookup tables (or neural networks) can store the mapping between participation factors and end-effector coordinates for a given set of manipulator modes. Interpolation (or neural network generalization) can be used to interpolate the data. Thus the speed associated with closed form inverse kinematic solutions can be attained for a wide variety of modes.

Alternatively, an approach analogous to the popular "resolved rate" method can be used. This scheme is based on the time derivative of the forward kinematic map of a backbone curve restricted to modal form:

$$\dot{\bar{x}} = \mathcal{T}(\bar{a})\dot{\bar{a}} \tag{25}$$

where $\mathcal{T}(\bar{a})$ is the modal Jacobian matrix. The components of this matrix are: $\mathcal{T}_{ik} = \partial x_i/\partial a_k$. If a closed form forward kinematic solution exists [such as the example in (22)], then the components of $\mathcal{T}$ can be simply computed. If only a numerical forward kinematic solution exists, the modal Jacobian matrix elements can be computed numerically by using Liebniz's rule. For example, the modal Jacobian elements for an inextensible planar backbone curve having modal shape function $\theta(s,t) = \Sigma_j^{N_\theta} a_j(t)\phi_j(s)$ can be numerically computed as:

$$\mathcal{T}_{1j} = \frac{\partial x_{ee}}{\partial a_j} = \frac{\partial}{\partial a_j}\int_0^1 \sin(\theta(\sigma,t)d\sigma = \int_0^1 \phi_j(\sigma)\cos(\theta(\sigma))d\sigma$$
$$\mathcal{T}_{2j} = \frac{\partial y_{ee}}{\partial a_j} = \frac{\partial}{\partial a_j}\int_0^1 \cos(\theta(\sigma,t)d\sigma = -\int_0^1 \phi_j(\sigma)\sin(\theta(\sigma))d\sigma. \tag{26}$$

(25) can be used in two ways. First, it can be used to determine the value of $\bar{a}$ that solves an inverse kinematic problem for fixed t. Let $\tilde{\bar{a}}_0(t_0)$ be an initial estimate of $\bar{a}$ at time t_0. (25) can be iterated in a differential form:

$$\tilde{\bar{a}}_{k+1}(t_0) = \tilde{\bar{a}}_k(t_0)+\alpha\mathcal{T}^{-1}(\tilde{\bar{a}}_k(t_0))[\bar{x}_D(t_0)-\bar{x}(\tilde{\bar{a}}_k(t_0))] \tag{27}$$

at fixed t_0. $\tilde{\bar{a}}_k$ is the estimated value of $\bar{a}$ at the kth iteration of (27). $\bar{x}(\tilde{\bar{a}}_k(t_0))$ is the backbone curve tip location computed using the estimated $\bar{a}$, while $\bar{x}_D(t_0)$ is the desired tip location. $\alpha > 0$ is a constant that controls the convergence rate of (27). For "small" α and "small" $\|\bar{x}_D(t_0) - \bar{x}(\tilde{\bar{a}}_k(t_0))\|$, (27) will converge to the proper modal participation factor. In fact, for small $\|\bar{x}_D(t_0) - \bar{x}(\tilde{\bar{a}}_k(t_0))\|$, the choice $\alpha = 1$ works well. Should $\|\bar{x}_D(t_0) - \bar{x}(\tilde{\bar{a}}_K(t_0))\|$ not be small, then a homotopy continuation technique can be used.

For example, consider the two mode example of (2). Assume a desired end-effector position of $\bar{x}_D = (0.2, 0.5)$. Set the initial values of the modal participation factors to $a_1 = 1, a_2 = 1$. Iteration of (27) with $\alpha = 1$ converges to the proper modal participation factors, $a_1 = 1.4011, a_2 = 0.38$, in 5 iterations (with end-effector error of less than 0.0001%).

If $\bar{x}_D$ is a function of time, then (25) can also be used for trajectory planning in a fashion analogous to the resolved rate method. Assuming that the $\bar{a}(t_0)$ is known at the beginning of the trajectory, then (25) can be numerically inverted and integrated along the path to find $\bar{a}(t)$. This is illustrated in Fig. 9 where the backbone curve tip follows a straight line, while the modes of (14) are employed.

D. Piecewise Continuous Modal Solutions

For some tasks, such as obstacle avoidance [2], it is convenient to define the backbone curve shape functions as the piecewise continuous sum of local shape functions. For

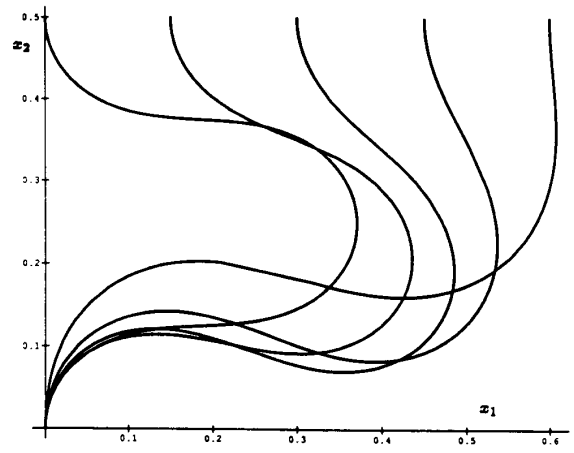


Fig. 9. Modal solutions for straight line trajectory.

example, in the planar nonextensible case, the shape function $\theta(s,t)$ can assume the form:

$$\theta(s,t) = \sum_{i=1}^{N_{seg}} \theta_i(s,t) W(s, s_{i-1}(t), s_i(t)). \tag{28}$$

where N_{seg} is the number of local backbone curve segments. In other words, one can define a "local" shape function, $\theta_i(s,t)$, which acts over the backbone curve segment $s_{i-1}(t) \leq s \leq s_i(t)$, where $s_0 = 0$ and $s_{N_{seg}} = 1$, and the segment endpoints may vary with time. The entire backbone curve shape function is a concatenation of the local shape functions. In this way, local constraints can be more naturally satisfied. Modal inverse kinematic techniques can then be defined for each segment. More detailed application of this approach to obstacle avoidance and hyper-redundant locomotion can be found in [2], [3].

V. Modal Singularities, Degenerate Modes, and Mode Switching

Algorithmic singularities will be introduced with any choice of a redundancy resolution scheme. Modal hyper-redundancy resolution has associated modal singularities. In a kinematic singularity, instantaneous joint motions can not produce end-effector motion in one or more directions. For modal singularities, the loss of end-effector freedom is measured with respect to instantaneous changes in modal participation factors. To illustrate the modal singularities, note that the modal Jacobian of the two-mode example (14) loses rank when:

$$det(\mathcal{T}) = \frac{a_1}{(a_1^2 + a_2^2)^{1/2}} J_1[(a_1^2 + a_2^2)^{1/2}] J_0[(a_1^2 + a_2^2)^{1/2}] = 0. \tag{29}$$

(29) will be satisfied when: $a_1 = 0; J_0[(a_1^2 + a_2^2)^{1/2}] = 0$; or $J_1[(a_1^2 + a_2^2)^{1/2}] = 0$. The case of $a_1 = 0$ corresponds to a workspace boundary imposed by these modes. The other conditions occur when $(a_1^2 + a_2^2)^{1/2}$ is a zero of J_0 or J_1. Since the inverse kinematics solution uses a restricted Bessel function, this occurs when $(a_1^2 + a_2^2)^{1/2} = \mu_1$, where $\mu_1 \simeq 2.405$ is the first zero of J_0. Fig. 10 shows the loci of backbone curve tip positions associated with these modal singularities.

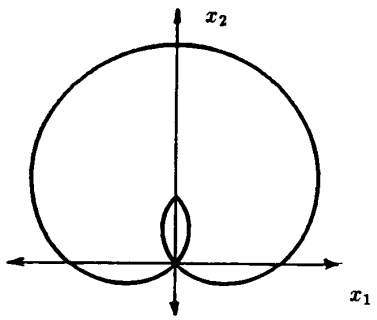


Fig. 10. Loci of modal singularities.

It should also be understood that the set of mode functions can not be chosen arbitrarily. First, the modes must be linearly independent on the interval $s \in [0,1]$. However, independency is not a sufficient criteria for the selection of "good" modal sets. Additional care must be exercised to avoid choosing degenerate modes.

Definition: A set of modes is degenerate if the modal Jacobian loses rank for all values of modal participation factors.

Here we illustrate this idea with a class of modes that are linearly independent, but are in fact degenerate. Consider an inextensible planar backbone curve where the shape function, $\theta(s,t)$, assumes a modal form. Let the set of mode functions, $\{\phi_i\}$, be antisymmetric, or odd, about the point $s = 1/2$. For example, $\phi_1(s) = \sin(2\pi s)$ and $\phi_2(s) = \sin(4\pi s)$ are linearly independent and odd on $s \in [0,1]$. Oddness implies that $\phi_i(s) = -\phi_i(1-s)$, which further implies that $\theta(s,t) = -\theta(1-s,t)$. The x_1-coordinate of the tip position is:

$$\begin{aligned} x_1(1,t) &= \int_0^1 \sin\theta(s,t) ds \\ &= \int_0^{1/2} \sin\theta(s,t) ds + \int_{1/2}^1 \sin\theta(s,t) ds \\ &= \int_0^{1/2} \sin\theta(s,t) ds - \int_0^{1/2} \sin\theta(\sigma,t) d\sigma = 0. \end{aligned} \tag{30}$$

where the change of variables $\sigma = 1 - s$ was introduced into the integral over the interval $s \in [1/2, 1]$. In other words, with this set of modes, the manipulator end-effector is constrained to move along the x_2-axis, even if the modes are linearly independent.

A. Switching Between Sets of Modes

Constraining the backbone curve's shape functions to a modal form may restrict the manipulator to operate in a workspace that is smaller than its physical capabilities. However, the physical workspace may be "covered" by different overlapping regions corresponding to each set of modes, as shown schematically in Fig. 11. In operation, the manipulator can switch from one mode set to the other. In addition, mode switching may also be desirable as the nature of the task changes and different mode functions are required.

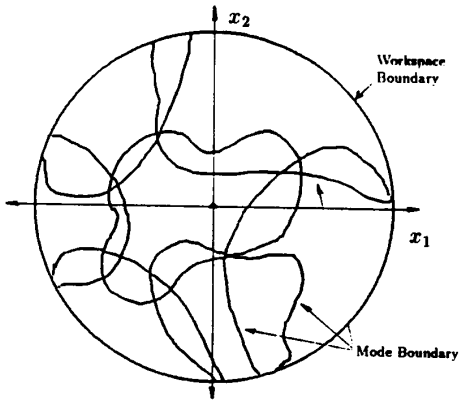


Fig. 11. Multiple mode covering of the workspace.

Assume that we wish to switch from an initial set of modes and participation factors $\{\phi_i^I, \overline{a}^I\}$ to the set $\{\phi_i^F, \overline{a}^F\}$. Let $f(t)$ be an increasing function such that $f(0) = 0$ and $f(1) = 1$ where $t = 0$ and $t = 1$ represent the beginning and end of the switching process. The switching process can be performed by letting each $\mathcal{S}_i(s,t)$ assume the following form during the switching phase:

$$\mathcal{S}_i(s,t) = (1 - f(t)) \sum_{i=1}^{N_{S_i}} a_i^I \phi_i^I + f(t) \sum_{i=1}^{N_{S_i}} a_i^F \phi_i^F \qquad (31)$$

The end-effector will not remain stationary during this mode switching. If desired, end-effector stationarity can be enforced as follows. Let $\mathcal{S}_i(s)$ be constructed from the sum of both sets of modes, $\phi_I(s)$ and $\phi_F(s)$. Thus, the time derivative of the forward kinematic map for a stationary end-effector is:

$$\dot{\overline{x}} = \mathcal{T}^I \dot{\overline{a}}^I + \mathcal{T}^F \dot{\overline{a}}^F = 0 \qquad (32)$$

where $\mathcal{T}^I$ and $\mathcal{T}^F$ are the modal Jacobians with respect to $\overline{a}^I$ and $\overline{a}^F$. At $t = 0$, let $\overline{a}^I(0)$ assume the values that place the end-effector at the desired stationary position. Define $\overline{a}^I(t) = \overline{a}^I(0)(1 - f(t))$, where $f(t)$ is an increasing function as above. At $t = 0$ let $\overline{a}^F(0) = 0$. (32) can be solved for $\dot{\overline{a}}^F(t)$, which can be numerically integrated to find $\overline{a}^F(t)$. In this way, the manipulator smoothly changes from the initial to final modes, while the end-effector remains stationary. If $\mathcal{T}^F$ loses rank, the switching can be computed backwards in time by defining $\overline{a}^F$ as a decreasing function, and solving for $\overline{a}^I(t)$. If both Jacobians are singular, the constraint of constant end-effector position during mode switching must be relaxed, or additional temporary modes must be introduced during the switching process.

VI. Fitting Algorithms for Discrete Morphologies

A backbone shape that satisfies task constraints may be derived from the modal method described above, an optimal approach [4], or other conceivable methods. In any case, a continuous backbone inverse kinematic solution will ultimately be used in practice to determine the robot actuator displacements that implement the desired motion. The continuous backbone curve solution can be used to directly specify the local deformation of a continuous morphology robot. This section considers "fitting" algorithms that adapt continuous curve solutions to discrete hyper-redundant robot morphologies. The actuator displacements are computed so that end-effector of the discrete manipulator geometry exactly matches the tip frame of the backbone curve, while the rest of the mechanism adheres "as closely as possible" to the continuous backbone curve shape. Different mechanical morphologies will require different fitting algorithms. We consider two discrete morphology classes to demonstrate different fitting styles. Fitting algorithms for other morphologies can be developed in an analogous fashion, and more examples can be found in [6].

A. Modular and VGT Morphologies

This section considers a fitting algorithm for manipulators with a "modular" structure, such as VGT robots. Assume that each module has the same or greater number of degrees of freedom as the ambient space (e.g., 3 for planar or 6 for spatial). Consider the ith module in a manipulator chain consisting of n modules. Attach a frame, $\{F_{i-1}\}$, to the "input", or base, of the ith module, and a frame, $\{F_i\}$, to the "output," or top, of the module. For the modular manipulator configuration to closely conform to the continuous curve geometry, the frames $\{F_{i-1}\}$ and $\{F_i\}$ are chosen to coincide with the backbone curve reference frames at "fitting" points s_{i-1} and $s_i, i = 1, \cdots, n$. The spacing of points $\{s_i\}$ is chosen uniformly: $s_i = i/n$ (note that [4] develops a method to optimally choose the fitting point spacing).

Let $\overline{r}_{i-1}^i$ denote the displacement and $[A]_{i-1}^i$ the orientation of $\{F_i\}$ with respect to $\{F_{i-1}\}$. It is assumed that the module inverse kinematics, which relates $\{F_i\}$ to $\{F_{i-1}\}$, can be solved (an example is given below). $\overline{r}_{i-1}^i$ and $[A]_{i-1}^i$ serve as the input to the inverse kinematic algorithm of each module. The discrete manipulator configuration will conform to the continuous backbone curve if:

$$\begin{aligned} [A]_{i-1}^i &= \mathbf{Q}^T(s_{i-1}\mathbf{Q}(s_i) \\ \overline{r}_{i-1}^i &= \mathbf{Q}^T(s_{i-1})(\overline{x}(s_i) - \overline{x}(s_{i-1})) \end{aligned} \qquad (33)$$

In this way, the frames fixed in the discrete manipulator, $\{F_i\}$, coincide exactly with the backbone curve reference frames at points $\{s_i\}$. Once the backbone curve inverse kinematic solution has been computed, each $[A]_{i-1}^i$ and $\overline{r}_{i-1}^i$ can be computed in parallel. Similarly, the module inverse kinematics can also be computed in parallel. Assuming a parallel computation structure of one processor per module, this method will require roughly the same computation time for manipulators with an arbitrary number of DOF.

Let us consider a planar VGT manipulator composed of modules having the geometry seen in Fig. 12 (this is the same geometry as the robot in Fig. 1). The module geometry is varied by adjusting the lengths of the module's three prismatic actuators. Let vectors $\overline{v}_0^i, \overline{v}_1^i$, and $\overline{c}^i$ be collinear with the axes of their respective prismatic actuators in the ith module (Fig. 12). These vectors are determined from the backbone curve

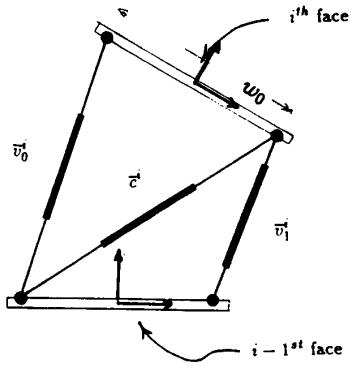


Fig. 12. One section of a planar VGT manipulator.

as follows:

$$\begin{aligned} \overline{v}_j^i &= \overline{r}_{i-1}^i - \overline{n}_j^i + [A]_{i-1}^i \overline{n}_j^i \quad j = 0,1 \\ \overline{c}^i &= \overline{r}_{i-1}^i - \overline{n}_1^i + [A]_{i-1}^i \overline{n}_2^i \end{aligned} \tag{34}$$

where $\overline{n}_1^i = [-w_0/2, 0]^T$ and $\overline{n}_2^i = [w_0/2, 0]^T$ where w_0 is the module width (Fig. 12). In the planar case:

$$\overline{r}_{i-1}^i = \begin{pmatrix} \int_{s_{i-1}}^{s_i} l(s)\sin[\theta(s) - \theta(s_{i-1})]ds \\ \int_{s_{i-1}}^{s_i} l(s)\cos[\theta(s) - \theta(s_{i-1})]ds \end{pmatrix}$$

$$[A]_{i-1}^i = \begin{bmatrix} \cos(\theta_i - \theta_{i-1}) & -\sin(\theta_i - \theta_{i-1}) \\ \sin(\theta_i - \theta_{i-1}) & \cos(\theta_i - \theta_{i-1} \end{bmatrix} \tag{35}$$

The module inverse kinematic solution is simply the lengths of these vectors, which correspond to the prismatic actuator displacements:

$$\lambda_1^i = \|\overline{v}_1^i\|; \qquad \lambda_2^i = \|\overline{v}_2^i\|; \qquad \lambda_3^i = \|\overline{c}^i\| \tag{36}$$

Fig. 13(a) shows a sequence of backbone curve configurations during an obstacle avoidance maneuver (the shape functions used for this example can be found in [2]). Fig. 13(b) illustrates a variable geometry truss "fitted" to some of the continuous backbone curves configurations using (34), (35), and (36).

The spatial VGT manipulator shown in Fig. 5 consists of a concatenation of Stewart platforms, and is highly analogous to the planar VGT of Fig. 13. An analogous procedure, which can be found in [6], was used to generate the configurations in Fig. 5 from a continuous backbone curve model.

B. Planar Revolute-Jointed Morphology

Consider the planar n-link manipulator shown in Fig. 2(a) with n large, but finite. The extensibility of the modules in Section VI-A allowed an exact fit between a finite number of points on the discrete manipulator and the continuous backbone curve. Because of the serial link mechanism's inextensibility, it can not be exactly matched to a specified set of backbone curve points. Instead, the mechanism is fitted to the backbone curve using a constrained least squares approach.

Assume that the link lengths are uniform, with value $1/n$. The first joint is located at $(x, y) = (0, 0)$, where for convenience we use the notation (x, y) instead of (x_1, x_2) to coordinatize the 2-dimensional plane. The forward kinematics of the manipulator is simply:

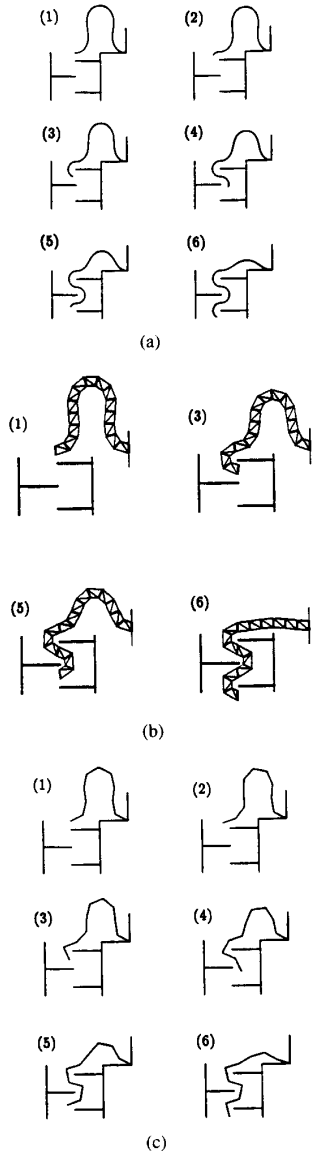


Fig. 13. Fitting of planar VGT and revolute jointed manipulator to a backbone curve solution.

$$x_{ee} = \frac{1}{n}\sum_{i=1}^{N} \sin q_i^*; \qquad y_{ee} = \frac{1}{n}\sum_{i=1}^{N} \cos q_i^*; \qquad \theta_{ee} = q_n^* \tag{37}$$

where x_{ee}, y_{ee}, and θ_{ee} are the position and orientation of the end-effector. q_i^* is the absolute angle of the ith link (which connects joints i and $i+1$) with respect to the y-axis. The ith joint angle, q_i, is the difference between adjacent absolute angles: $q_i = q_i^* - q_{i-1}^*$.

The backbone curve associated to this manipulator is assumed to be inextensible. Further assume that a continuous backbone curve inverse kinematic solution, $\theta(s)$, has been

computed at fixed t. A "fitting error function" is defined as follows:

$$G = \frac{1}{2}\sum_{i=1}^{n}({}^{c}x(s_i) - {}^{d}x_i)^2 + ({}^{c}y(s_i) - {}^{d}y_i)^2 \qquad (38)$$

where ${}^{c}x(s_i)$ and ${}^{c}y(s_i)$ are respectively the x and y-position of the continuous backbone curve solution at points $s_i = i/n$ for $i = 1, \cdots, n$. ${}^{d}x_i$ and ${}^{d}y_i$ are respectively the x and y-location of points on the discrete manipulator that are to be closely matched to points ${}^{c}x_i$ and ${}^{c}y_i$. Physically, G is the sum of the squared error between the continuous backbone curve fitting points and the associated points on the discrete manipulator. Here we choose $({}^{d}x_i, {}^{d}y_i)$ to be the location of joint $(i+1)$, or the end-effector, if $i = n$:

$${}^{d}x_i = \frac{1}{n}\sum_{j=1}^{i}\sin q_j^*; \qquad {}^{d}y_i = \frac{1}{n}\sum_{j=1}^{i}\cos q_j^*. \qquad (39)$$

Fitting then reduces to the minimization of (38) with respect to the $\{q_i^*\}$. This minimization must additionally be constrained so that the discrete and continuous end-effectors coincide exactly. Let the end-effector constraints be denoted: $g_1 = {}^{c}x_1(1) - {}^{d}x_n = 0; g_2 = {}^{c}y_1(1) - {}^{d}y_n = 0; g_3 = \theta_{ee} - q_n^* = 0$.

While any constrained numerical optimization technique can be employed, an efficient linearized optimization scheme can be realized as follows. First, we estimate q_j^*. Assuming large n, the jth absolute link angle might be approximated by the angle of the tangent to the backbone curve at $s_j : \tilde{q}_j^* = \theta(s_i)$. Or, one might choose slightly more complicated (and accurate) estimates, such as $\cos \tilde{q}_j^* = [{}^{c}\overline{x}(s_{i-1}) - {}^{c}\overline{x}(s_i)] \cdot [{}^{c}\overline{x}(s_{i+1}) - {}^{c}\overline{x}(s_i)]$. However, we know that this approximation will be in error, and so we introduce n free "fitting" parameters, ϵ_j:

$$q_j^* = \tilde{q}_j^* + \epsilon_j \qquad (40)$$

where ϵ_j is the error in the jth joint angle between the best fit and the approximate fit. Since the estimate is assumed to be good, and thus the $\{|\epsilon_j|\}$ are small, we can linearize the fitting error and the end-effector position constraint equations with respect to the $\{\epsilon_j\}$. Substituting (40) into (38) and linearizing about small $\{\epsilon_j\}$ results in:

$$G \approx \frac{1}{2}\sum_{j=1}^{n}\left({}^{c}x_j - \frac{1}{n}\sum_{i=1}^{j}\sin\tilde{q}_i^* + \epsilon_i\cos\tilde{q}_i^*\right)^2 + \left({}^{c}y_i - \frac{1}{n}\sum_{i=1}^{j}\cos\tilde{q}_i^* - \epsilon_i\sin\tilde{q}_i^*\right)^2 \qquad (41)$$

The linearized end-effector position constraints are:

$$g_1 = {}^{c}x(1) - {}^{d}x_n \simeq x_{ee} - \frac{1}{n}\sum_{j=1}^{n}\sin\tilde{q}_j^* + \epsilon_j\cos\tilde{q}_j^* = 0$$

$$g_2 = {}^{c}y(1) - {}^{d}y_n \simeq y_{ee} - \frac{1}{n}\sum_{j=1}^{n}\cos\tilde{q}_j^* - \epsilon_j\sin\tilde{q}_j^* = 0$$

$$g_3 = \theta_{ee} - (q_n^* + \epsilon_n) = 0 \qquad (42)$$

A necessary condition for the constrained minima of (41) is:

$$\nabla_\epsilon G = \sum_{i=1}^{3}\lambda_i\nabla_\epsilon g_i. \qquad (43)$$

(43) and the constraint equations (42) provide $n+3$ linear equations in the n variable $\{\epsilon_j\}$ and three Lagrange multipliers $\{\lambda_i\}$. These can be solved for the values of $\{\epsilon_j\}$ that minimize G. If the calculated $\{\epsilon_i\}$ are "small," the linearization assumptions are valid. If the $\{\epsilon_i\}$ are not sufficiently small, the above procedure can be iterated a few times, or a more complicated minimization procedure must be implemented. This will only arise when the backbone curvature is very large. Fig. 13 shows an example in which a 10 link revolute-jointed robot is fitted to the example of Fig. 13(a).

VII. Conclusions

This paper developed a novel approach to hyper-redundant manipulator kinematics. In this approach, hyper-redundant robots are abstractly represented by backbone curves that either exactly or closely capture the important macroscopic geometric features of the robot. Hyper-redundancy resolution reduces to determining the proper time varying behavior of the backbone curve. Novel alternatives to the classical Frenet-Serret parametrization were introduced to parametrize the backbone curve. These parametrization techniques are general, physically motivated, and are useful regardless of the particular hyper-redundancy resolution method that is used.

The continuous backbone curve shape can be used directly to specify the local bending and twisting of a continuous morphology manipulator. For discretely segmented or modular hyper-redundant morphologies, we introduced "fitting" procedures to determine discrete actuator displacements from the continuous solutions. Thus, the continuous kinematic analysis is applicable to all hyper-redundant manipulator morphologies, with an additional step required to specialize the computations for a particular mechanical structure.

A hyper-redundant robot user must select some criteria for hyper-redundancy resolution. This paper focused on a modal hyper-redundancy resolution scheme that constrained the backbone curve shape functions to a modal form. This approach can lead to highly efficient inverse kinematic solutions. Additionally, when the number of modes is equal to the number of task constraints, the modal method is cyclic in workspace regions free of modal singularities Further, as shown in [2], [3], the modal method is well suited to many hyper-redundant robot operations, such as locomotion or obstacle avoidance. References [2] and [3] also show how physical task characteristics can be used to select mode shapes. This paper focused on the basic backbone modeling technique, and some of details, such as singularities and degenerate modes, of the modal inverse kinematic approach. The modal approach is only one method for finding backbone curve shapes that satisfy task constraints. In [4] we considered techniques to find the optimal backbone curve shape. The techniques are much less computationally tractable, and thus the modal approach has many advantages for practical applications.

Hyper-redundant robots have failed to achieve widespread applicability, due in part to the inefficiency and ineffectiveness of previous kinematic modeling techniques. The algorithms developed in this paper are a step toward efficient kinematic control of hyper-redundant robots. We have implemented these algorithms on the 30 DOF robot shown in Fig. 1, where only a single Motorola 68030 processor is required to both servo all 30 actuators and compute the modal inverse kinematic solution. References [5] and [7] should be consulted for details of this implementation and experimental results obtained with this system.

REFERENCES

[1] G. S. Chirikjian and J. W. Burdick, "Parallel formulation of the inverse kinematics of modular hyper-redundant manipulators," in *Proc. IEEE Int. Conf. Robotics Automat.*, Sacramento, CA, April, 1991.

[2] G. S. Chirikjian and J. W. Burdick, "An obstacle avoidance algorithm for hyper-redundant manipulators," in *Proc. IEEE Int. Conf. Robotics Automat.*, Cincinnati, OH, May 14–17, 1990.

[3] G. S. Chirikjian and J. W. Burdick, "Kinematics of hyper-redundant robot locomotion with applications to grasping," in *Proc. IEEE Int. Conf. Robotics Automat.*, Sacramento, CA, April 1991.

[4] G. S. Chirikjian and J. W. Burdick, "On the determination of kinematically optimal hyper-redundant manipulator configurations" in *Proc. IEEE Int. Conf. Robotics Automat.*, Nice, France, May 10–15, 1992.

[5] G. S. Chirikjian and J. W. Burdick, "Design, implementation, and experiments with a thirty degree-of-freedom 'hyper-redundant' robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, Atlanta, GA, 1993.

[6] G. S. Chirikjian, "Theory and applications of hyper-redundant robotic manipulators," Ph.D. thesis, Dept, of Applied Mechanics, California Institute of Technology, Pasadena, CA, June, 1992.

[7] G. S. Chirikjian and J. W. Burdick, "Experiments in hyper-redundant manipulation," to appear in *Video Proceedings, IEEE Int. Conf. Robotics Automat.*, Atlanta, GA, 1993.

[8] A. Hayashi, J. Park, and B. J. Kuipers, "Toward planning and control of highly redundant manipulators," in *Fifth IEEE Int. Symp. on Intelligent Control, 1990.*

[9] M. Ivanescu and I. Badea, "Dynamic control for a tentacle manipulator," in *Proc. Int. Conf. on Robotics and Factories of the Future*, Dec. 4–7, 1984, Charlotte, NC, USA, pp. 317–328.

[10] J. S. Pettinato and H. E. Stephanou, "Manipulability and stability of a tentacle based robot manipulator," in *Proc. IEEE Int. Conf. Robotics Automat.*, Scottsdale, AZ, May 1989, pp. 458–463.

[11] F. Naccarato and P. C. Hughes, "An inverse kinematics algorithm for a highly redundant variable-geometry-truss manipulator," in *Proc. 3rd Annual Conf. Aerospace Computational Control*, D. E. Bernard and G. K. Man, Eds. Oxnard, CA: JPL Publication, 89–45, Dec. 15, 1989.

[12] A. Morecki, *et al.*, "Robotic system—Elephant trunk type elastic manipulator combined with a quadruped walking machine," in *Proc. of Second Int. Conf. on Robotics and Factories of the Future*, San Diego, July 1987, pp. 649–656.

[13] S. Hirose and Y. Umetani, "Kinematic control of active cord mechanism with tactile sensors," in *Proc. 2nd Int. CISM-IFT Symp. on Theory and Practice of Robots and Manipulators*, pp. 241–252, 1976.

[14] S. Hirose and A. Morishima, "Design and control of a mobile robot with an articulated body," *Int. J. Robotics Res.*, vol. 9, no. 2, pp. 99–114, 1990.

[15] D. Tesar and M. S. Butler, "A generalized modular architecture for robot structures," *ASME Manufacturing Review*, vol. 2, no. 2, pp. 91–118, June 1989.

[16] T. S. Drozda, "Spine robot...The verdict's yet to come," *Manufacturing Engineering*, vol. 93, no. 3, pp. 110–112, 1984.

[17] V. V. Anderson and R. C. Horn, "Tensor-arm Manipulator design," *ASME Trans.*, vol. 67-DE-57, pp. 1–12, 1967.

[18] J. F. Wilson and U. Mahajan, "The mechanics and positioning of highly flexible manipulator limbs," *ASME J. of Mech., Trans., Automat. Design*, vol. 111, June 1989.

[19] R. J. Salerno, C. F. Reinholtz, and H. H. Robertshaw, "Shape control of high degree-of-freedom variable geometry trusses," in *Proc. Workshop on Computational Aspects in the Control of Flexible Systems, Part 2*, Williamsburg, VA, July 12–14, 1988.

[20] S. Tavakkoli and S. G. Dhande, "Shape synthesis and optimization using intrinsic geometry," in *Proc. ASME Design Conference*, Chicago, IL, Sept. 16–19, 1990.

[21] T. Fukuda, H. Hosokai, and I. Kikuchi, "Distributed type of actuators by shape memory alloy and its application to underwater mobile robotics mechanism," in *Proc. IEEE Int. Conf. on Robotics Automat.*, Cincinnati, OH, May 14–17, 1990, pp. 1316–1321.

[22] A. Hemami, "Studies on a light weight and flexible robot manipulator," *Robotics*, vol. 1, pp. 27–36, 1985.

[23] C. A. Klein and C. H. Huang, "Review of the pseudoinverse for control of kinematically redundant manipulators," *IEEE Trans. Syst. Man Cyber.*, March, 1983.

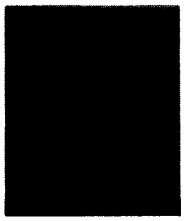
[24] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[25] F. Bowman, *Introduction to Bessel Functions*. New York: Dover Inc., 1958.

[26] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometry Design*. New York: Springer-Verlag, 1988.

[27] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. of the IEEE Intern. Conf. Robotics and Automat.*, St. Louis, MO, March 25–28, 1985, pp. 722–728.

[28] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. Robotics Automat.*, vol. 5, no. 4, pp. 472–490, 1989.



Gregory S. Chirikjian received the B.S.E. degree in Engineering Mechanics and the M.S.E. degree in Mechanical Engineering in 1988 while also fulfilling requirement for a B.A. in Mathematics, all from The Johns Hopkins University. Between 1988 and 1992 he was a graduate student at the California Institute of Technology, and in 1992 he received the Ph.D. degree in Applied Mechanics from the California Institute of Technology. Since the summer of 1992 he has been an Assistant Professor in the Department of Mechanical Engineering at Johns Hopkins, and started the robotics program there. His general research interests are in the kinematic analysis, design, and implementation of "hyper-redundant," "metamorphic," and "binary" manipulators. He recently received an NSF New Young Investigator award to investigate these areas.

Joel. W. Burdick received the B.S.M.E. degree from Duke University in 1981, and the Ph.D. degree in mechanical engineering from Stamford University in 1988. Since May, 1988 he has been an assistant professor in the Department of Mechanical Engineering at the California Institute of Technology. He is a National Science Foundation Presidential Young Investigator, an Office of Naval Research Young Investigator, and the Hughes Feynman Fellow. His general research interests are in hyper-redundant robots, medical robots, dynamically stable legged locomotion, and kinematic mobility theory.