

Hyper-redundant manipulator dynamics: a continuum approximation

GREGORY S. CHIRIKJIAN

Department of Mechanical Engineering, 124 Latrobe Hall, Johns Hopkins University, Baltimore, MD 21218, USA

Received for AR 17 November 1993

Abstract—Hyper-redundant, or ‘snake-like’, manipulators have a very large number of actuatable degrees of freedom. This paper develops an efficient formulation of approximate hyper-redundant manipulator dynamics. The most efficient methods for representing manipulator dynamics in the literature require serial computations proportional to the number of degrees of freedom. Furthermore, these methods are not fully parallelizable. For hyper-redundant manipulators, which may have tens, hundreds or thousands of actuators, these formulations preclude real time implementation. This paper therefore looks at the mechanics of hyper-redundant manipulators from the point of view of an approximation to an ‘infinite degree-of-freedom’ (or continuum) problem. The dynamics for this case is developed. The approximate dynamics of actual hyper-redundant manipulators is then reduced to a problem which is $O(1)$ in time, i.e. the algorithm is $O(n)$ is the total number of computations, but these computations can be completely distributed over n parallel processors. This is achieved by ‘projecting’ the dynamics of the continuum model onto the actual robotic structure. Applications of this method to practical computed torque control schemes for hyper-redundant manipulators is demonstrated with two examples: (i) industrial pick-and-place tasks and (ii) inspection in an environment filled with viscous sludge, such as a hazardous waste dump. The results are compared with a lumped mass model of a particular hyper-redundant manipulator.

1. INTRODUCTION

Hyper-redundant manipulators have a very large number of actuatable degrees of freedom. Applications of ‘snake-like’ hyper-redundant manipulators include inspection in highly constrained environments, tentacle-like grasping of objects and whole-arm manipulation. Snake-like devices also have applications to locomotion [1–3].

Computationally attractive modeling of the system kinematics and dynamics is necessary for hyper-redundant manipulators to be used effectively. An efficient framework for the kinematics and motion planning of hyper-redundant manipulators is developed in [4, 12], and demonstrated in hardware in [5]. That framework is based on a continuous curve (or ‘continuum’) approximation which captures the manipulator’s macroscopic geometric features. The continuum approach is an alternative to methods developed recently for particular hyper-redundant robot morphologies [6, 7, 19].

This paper extends the continuum approach previously used for hyper-redundant manipulator kinematics to include efficient formulation of approximate hyper-redundant manipulator dynamics. The most efficient methods for representing manipulator dynamics in the literature require serial computations which grow linearly with the number of degrees of freedom [8, 9]. Furthermore, these methods are not fully parallelizable because serial iterations in force and velocity are intrinsic to their nature. For hyper-redundant manipulators, which may have tens, hundreds or thousands of actuators, this is not acceptable. This paper therefore looks at the dynamics problem for hyper-redundant manipulators from the point of view of an approximation to an ‘infinite degree-of-freedom’ problem. The dynamic equations for this infinite degree-of-freedom continuum model are developed. The dynamics of the continuum model is then ‘projected’ onto actual robotic structures. Application of this method to practical computed torque control schemes for hyper-redundant manipulators is demonstrated and compared with a lumped mass model.

This paper is organized as follows: Section 2 reviews previous formulations of robotic manipulator dynamics, basic principles of continuum mechanics and the kinematics of hyper-redundant manipulators. Section 3 uses the principles of continuum mechanics to approximately represent the dynamics of hyper-redundant manipulators. Section 3 also defines a procedure for ‘projecting’ the dynamics of the continuum model onto actual robotic structures. This approach is demonstrated with closed form solutions applied to a specific manipulator morphology: the Variable Geometry Truss (VGT) manipulator. Section 4 applies this new dynamics algorithm to two problems: (i) industrial pick-and-place tasks and (ii) inspection in a hazardous waste dump. Section 5 compares the new algorithm with existing methods.

2. BACKGROUND AND REVIEW

This section contains a review of a broad selection of material. Subsection 2.1 briefly reviews standard techniques for formulating the dynamics of robotic manipulators. Subsection 2.2 reviews some basic laws in continuum mechanics — an area of mechanics not commonly used in robotics. Subsection 2.3 reviews the author’s previous techniques for describing hyper-redundant manipulator kinematics.

2.1. Standard formulations of manipulator dynamics

The manipulator dynamics problem is generally formulated using techniques from Lagrangian mechanics or iterative Newton–Euler formulations. Lagrangian mechanics results in equations of motion of the form:

$$\mathbf{M}(\vec{q})\ddot{\vec{q}} + \vec{C}(\vec{q}, \dot{\vec{q}}) + \vec{G}(\vec{q}) = \vec{\tau}. \quad (1)$$

The direct evaluation of the left-hand side of the above dynamical equations for a given trajectory in joint space, $\vec{q}(t) \in \mathbb{R}^N$, requires $O(N^3)$ computations for a manipulator with N d.o.f. This is often referred to as the ‘inverse dynamics’ problem

[10]. It has been shown that Lagrangian formulations can be improved so as to have greater computational efficiency [8]. Nonetheless, the most commonly used method for formulating manipulator dynamics efficiently is the iterative Newton–Euler technique [9].

In the iterative Newton–Euler method, serial iterations in velocity are propagated forward from the manipulator base to the end-effector and forces are propagated backwards from the end-effector to the base. The equations associated with this procedure are given in [9]. The Lagrangian and iterative Newton–Euler formulation produce the same results. That is, torques are computed based on the desired joint trajectory. The difference is that the Lagrangian technique computes the vector of torques, and the iterative Newton–Euler approach generates components of the torque vector sequentially.

Computational aspects of these, and other, methods of formulating manipulator dynamics can be summarized by simply stating that the best methods require $O(N)$ serial computations. As one might expect, even this can become a heavy computational burden when considering hyper-redundant manipulators, where the number of degrees of freedom may be on the order of dozens or even hundreds. For this reason, it is worth investigating ‘continuum’ approximations to hyper-redundant manipulator dynamics.

2.2. Review of continuum mechanics

Continuum mechanics is a field of study concerned with the kinematics and dynamics of deformable media [11, 18]. Suppose we are given a deformable object. Let \vec{x} denote the position of an infinitesimal particle within this object. The mass of the object is given by

$$M = \int_V \rho \, dV,$$

where $\rho = \rho(\vec{x}, t)$ is the mass density of the object and V is the volume occupied by the object. Similarly, the linear momentum of the object is given by

$$\vec{P} = \int_V \rho \vec{v} \, dV,$$

where $\vec{v} = d\vec{x}/dt$, and the angular momentum of the object is

$$\vec{L} = \int_V \vec{x} \times \rho \vec{v} \, dV.$$

Conservation of mass is therefore written as $dM/dt = 0$, the momentum balance is $d\vec{P}/dt = \vec{F}$ and the angular momentum balance is $d\vec{L}/dt = \vec{N}$, where \vec{F} and \vec{N} are, respectively, the force and moment applied to the object. In the case of a continuum, these are broken down into contributions at the surface of the object and those which act through the volume. The former are called ‘tractions’ and the latter are called ‘body forces’. In addition, these balances apply to any subdivision of the object.

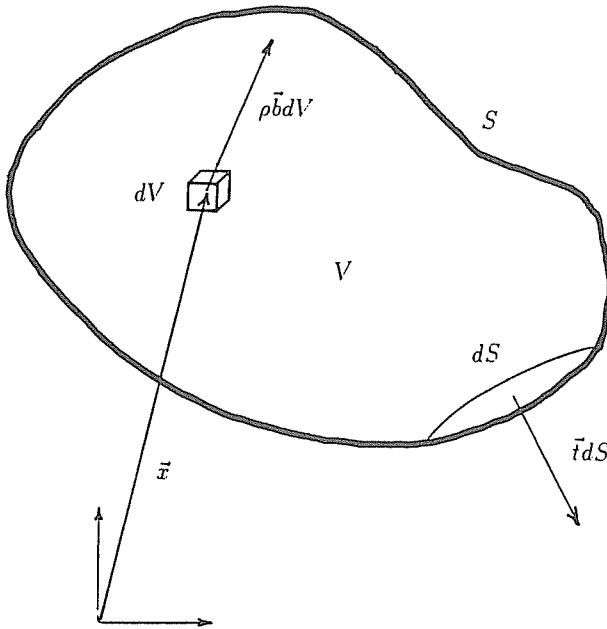


Figure 1. Forces acting on a continuum.

Figure 1 shows the body forces acting on an arbitrary cube within the volume and the applied surface tractions. Using a few classical arguments, the above laws: (i) mass balance, (ii) momentum balance and (iii) angular momentum balance are written in control volume form, respectively, as:

$$\frac{d}{dt} \int_V \rho dV = \frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \vec{v} \cdot \vec{n} dS = 0, \quad (2)$$

$$\int_S \vec{t} dS + \int_V \rho \vec{b} dV = \frac{d}{dt} \int_V \rho \vec{v} dV, \quad (3)$$

$$\int_S (\vec{x} \times \vec{t}) dS + \int_V (\vec{x} \times \rho \vec{b}) dV = \frac{d}{dt} \int_V (\vec{x} \times \rho \vec{v}) dV. \quad (4)$$

The subscripts S and V denote integrals over surface and volume of the region under consideration. \vec{n} is the normal to the control volume. \vec{t} is the applied surface 'traction'. \vec{b} is the body force per unit mass acting on the volume (e.g. gravity, magnetism, etc.).

It is interesting and useful for the formulation in Section 3 to note that the conservation of mass equation can be written different as:

$$\int_V \rho(\vec{x}, t) dV = \int_{V_0} \rho_0(\vec{X}) dV_0 \quad (5)$$

where \vec{X} is the position of a given infinitesimal volume (or particle) at time $t = 0$ and $\vec{x} = \vec{x}(\vec{X}, t)$ is the position of that particle at all future times, i.e. $\vec{x}(\vec{X}, 0) = \vec{X}$. V_0 denotes the volume occupied by the object at time $t = 0$, i.e. $dV_0 = dX_1 dX_2 dX_3$, whereas $dV = dx_1 dx_2 dx_3$. The integrals above are taken over the same material particles, although the size and shape of the volume may change.

By changing the domain of integration of the expression on the left side of equation (5), we get:

$$\int_V \rho(\vec{x}, t) dV = \int_{V_0} \rho(\vec{x}(\vec{X}, t), t) J(\vec{X}, t) dV_0, \quad (6)$$

where $J(\vec{X}, t) = \det(\partial \vec{x} / \partial \vec{X})$ is the determinant of the Jacobian matrix of the transformation $\vec{x} = \vec{x}(\vec{X}, t)$. Clearly, since both expressions hold for all possible volumes, we get

$$\rho_0(\vec{X}) = \rho(\vec{x}(\vec{X}, t), t) J(\vec{X}, t) \quad \text{or} \quad \rho(\vec{x}(\vec{X}, t), t) = \rho_0(\vec{X}) / J(\vec{X}, t). \quad (7)$$

In other words, by knowing the density of the object at time zero and knowing how it has deformed from that state, we know what its density is at the current time. This will be particularly useful for hyper-redundant manipulators that stretch and contract, thus changing density per unit length.

The next subsection reviews hyper-redundant manipulator kinematics, which forms the foundation for a continuum model of hyper-redundant manipulator dynamics.

2.3. Kinematics of backbone curves

It is assumed here that regardless of mechanical implementation, the important macroscopic features of a hyper-redundant robotic manipulator can be captured by a *backbone curve* and associated set of reference frames which evolve along the curve. A backbone curve parametrization and set of reference frames are collectively referred to as the *backbone reference set*. In this formulation, inverse kinematics and trajectory planning tasks are reduced to the determination of the proper time varying behavior of the backbone reference set [4, 12]. Note that depending on the actual mechanical implementation of the robot, the associated backbone curve may be *inextensible* (fixed length) or *extensible* (variable length).

A continuous backbone curve inverse kinematic solution (which may be generated by a ‘modal approach’ [12], ‘optimality-based approach’ [13] or any other method) can be used to directly determine the actuator displacements of a continuous morphology robot, e.g. such as one constructed from pneumatic actuator bundles. For discretely segmented modular morphologies, such as the one shown in Fig. 2, the continuous curve solution can be used, via a ‘fitting’ procedure [4, 12], to compute the actuator displacements which cause the manipulator to assume the nominal shape of the backbone curve model. In other words, the actual manipulator configuration is ‘algorithmically linked’ to the backbone curve model.

Techniques for the physically meaningful parametrization of backbone reference sets are reviewed in Subsection 2.3.1. Subsection 2.3.2 reviews how actual manipulators

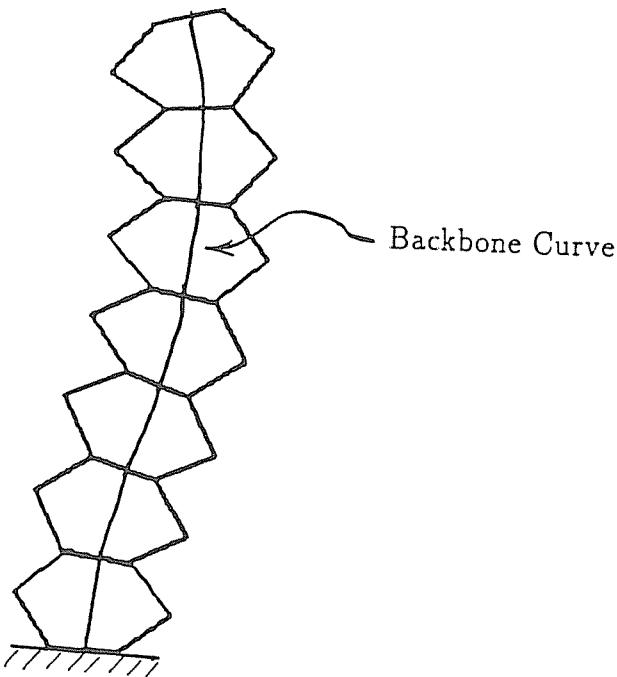


Figure 2. A modular morphology with superimposed backbone curve.

are ‘fit’ to the curve model. For the sake of brevity, only planar examples are used to illustrate these concepts. For more general formulations consult [4, 12].

2.3.1. Kinematics of backbone reference sets. The position of points on a backbone curve can be parametrized in the form:

$$\vec{x}(s, t) = \int_0^s [1 + \epsilon(\sigma, t)] \vec{u}_2(\sigma, t) d\sigma, \quad (8)$$

where $s \in [0, 1]$ is a parameter measuring distance along the backbone curve at time t . s need not be the classical arc length, which is denoted below as L . $\vec{x}(s, t)$ is a position vector from the base of the backbone curve to the point on the backbone curve denoted by curve parameter s . $\vec{u}_2(s, t)$ is the unit tangent vector to the curve at s . $\epsilon(s, t)$ is the *local extensibility* of the manipulator. $\epsilon(s, t)$ physically expresses how the backbone curve, which abstractly represents important geometric aspects of the real robot, locally expands or contracts relative to a given reference state, or ‘home’ configuration, of the robot. $\epsilon(s, t) > 0$ indicates local extension, while $\epsilon(s, t) < 0$ implies local contraction. One can also interpret the extensibility as a measure of how the parameter s differs from dimensionless arc length by computing arc length in the regular way [14]:

$$L(s, t) = \int_0^s [1 + \epsilon(\sigma, t)] d\sigma. \quad (9)$$

Using localization arguments, it is clear that the only time s is equal to L is when $\epsilon(s, t) = 0$. It will be useful to assume that there is a time, say $t = 0$, when $\epsilon(s, 0) = 0$. This will be called the reference state and in this state $L(s, 0) = s$. This fact will be applied to hyper-redundant manipulator dynamics in the same way that $\vec{x}(\vec{X}, 0) = \vec{X}$ is applied in general continuum mechanics. For compactness of notation, the following is defined:

$$l(s, t) = \partial L / \partial s = 1 + \epsilon(s, t).$$

The parametrization of equation (8) has the following interpretation. The backbone curve is 'grown' from the base by propagating the curve forward along the tangent vector, which is varying its direction according to $\vec{u}_2(s, t)$ and varying its magnitude (or 'growthrate') according to $l(s, t)$.

In the planar case, the locus of backbone curve points is defined by $\vec{x}(s, t) = [x_1(s, t), x_2(s, t)]^T$, where

$$x_1(s, t) = \int_0^s l(\sigma, t) \sin \theta(\sigma, t) d\sigma, \quad (10)$$

$$x_2(s, t) = \int_0^s l(\sigma, t) \cos \theta(\sigma, t) d\sigma. \quad (11)$$

$\theta(s, t)$ is the clockwise measured angle which the tangent to the curve at point s makes with the x_2 -axis at time t . Figure 3(b) illustrates the physical meaning of $l(s, t)$ and $\theta(s, t)$. A simple relationship exists between the classical curvature function of the curve and the functions θ , and l :

$$\kappa = \frac{\partial \theta}{\partial L} = \frac{1}{l} \frac{\partial \theta}{\partial s}.$$

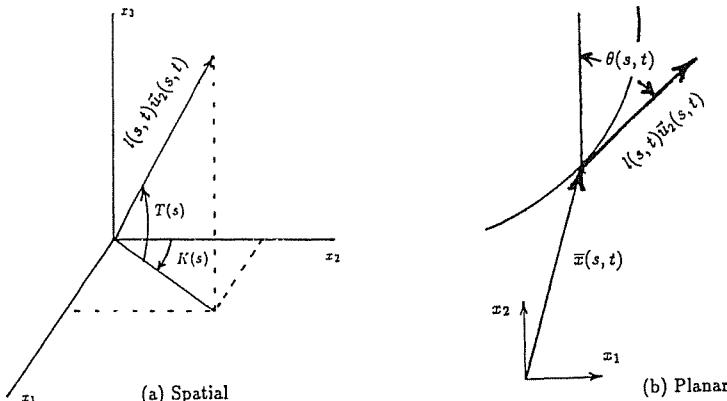


Figure 3. Description of backbone curve parametrization

A frame can be assigned to every point on a planar curve defined by $\theta(s, t)$. This frame is denoted by

$$\mathbf{Q}(s, t) = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}. \quad (12)$$

A similar formulation results for the spatial case, where angles denoted K and T define the direction of \vec{u}_2 (see Fig. 3a), and a roll about the backbone curve completes the description.

For consistency, the second column of \mathbf{Q} is chosen to be the backbone curve tangent vector, \vec{u}_2 .

All the information contained in a planar backbone curve model is conveniently expressed as a parametrized set of homogeneous transforms:

$$\mathcal{H}(s, t) = \begin{pmatrix} \mathbf{Q}(s, t) & \vec{x}(s, t) \\ \vec{0}^T & 1 \end{pmatrix}. \quad (13)$$

In summary, the kinematics of a backbone reference set, which uniquely describes a hyper-redundant manipulator backbone curve configuration, can be described by a set of physically meaningful geometric functions, which in the planar case are $\theta(s, t)$, and $\epsilon(s, t)$. This general formulation contrasts recent work in which a continuous curve model was used for the kinematics of inextensible revolute-jointed kinematic chains [15].

A ‘fitting’ procedure uses the set of frames defined by $\mathcal{H}(s, t)$ to cause a hyper-redundant manipulator to adhere to the backbone curve. Thus, the curve together with a set of reference frames and a fitting procedure define the macroscopic geometry of the manipulator. The next subsection details a parallelizable fitting procedure.

2.3.2. Inverse kinematics in parallel via ‘fitting’. A parallel algorithm developed in [12], which is based on the formulation of the previous subsubsection, is reviewed here. Manipulators with a modular architecture are considered. For example, the modules of an extensible spatial hyper-redundant manipulator might be Stewart platforms. It is assumed for simplicity that the modules are uniform in structure and size.

The backbone reference set can be used to generate inverse kinematic solutions for modular manipulators as follows. Consider the i th module in the manipulator chain consisting of n modules. Attach a frame (represented as a homogenous transform), \mathbf{H}^{i-1} , to the ‘input’, or base, of the module, and a frame, \mathbf{H}^i , to the ‘output’, or top, of the module. For the discretely segmented modular manipulator configuration to conform to the continuous curve geometry, the frames \mathbf{H}^{i-1} and \mathbf{H}^i are chosen to coincide with the backbone reference frames at points given by $s = (i-1)/n$ and $s = i/n$, respectively (see Fig. 4). That is, equate \mathbf{H}^i to $\mathcal{H}(i/n, t)$, which was defined in equation (13). Recall that equal partitioning of the curve parameter need not imply equal spacing along the curve because in general $L(s, t) \neq s$.

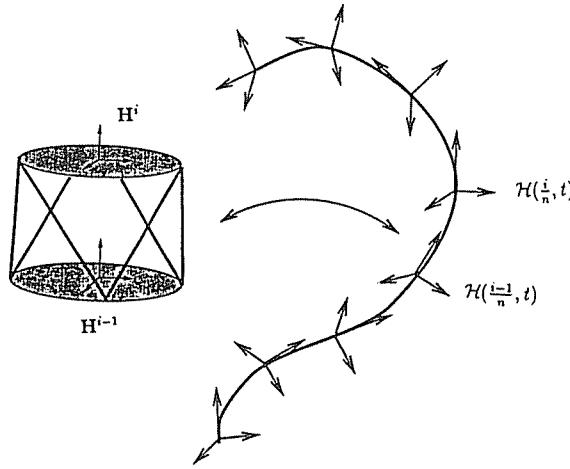


Figure 4. Fitting manipulator modules to the backbone

The frame \mathbf{H}^i measured relative to frame \mathbf{H}^{i-1} is denoted \mathbf{H}_{i-1}^i . This consists of the relative translation, \vec{r}_{i-1}^i , and rotation, \mathbf{R}_{i-1}^i :

$$\mathbf{H}_{i-1}^i(\vec{q}^{M_i}) = \begin{pmatrix} \mathbf{R}_{i-1}^i(\vec{q}^{M_i}) & \vec{r}_{i-1}^i(\vec{q}^{M_i}) \\ \vec{0}^T & 1 \end{pmatrix}. \quad (14)$$

$\vec{q}^{M_i} \in \mathbb{R}^m$ is the vector of joint displacements which determine the geometry of the i th module. In the plane, $m = 3$, while in space $m = 6$. The total number of manipulator degrees of freedom is then $N = nm$. It is assumed that the inverse kinematics of each module, which relates \mathbf{H}^i to \mathbf{H}^{i-1} , can be solved in a closed or efficient form (which is commonly the case for platform manipulator modules) for $i \in [1, \dots, n]$.

The manipulator configuration will conform to the backbone reference set if:

$$\mathbf{H}_{i-1}^i(\vec{q}^{M_i}(t)) = \mathcal{H}^{-1}\left(\frac{i-1}{n}, t\right) \mathcal{H}\left(\frac{i}{n}, t\right). \quad (15)$$

In the planar case, equation (15) is rewritten as:

$$\mathbf{R}_{i-1}^i(\vec{q}^{M_i}) = \mathbf{P}_{i-1}^i(t) \quad (16)$$

and

$$\vec{r}_{i-1}^i(\vec{q}^{M_i}) = \vec{p}_{i-1}^i(t), \quad (17)$$

where

$$\vec{p}_{i-1}^i = \left(\begin{array}{l} \int_{i-1/n}^{i/n} l(s, t) \sin [\theta(s, t) - \theta(\frac{i-1}{n}, t)] ds \\ \int_{i-1/n}^{i/n} l(s, t) \cos [\theta(s, t) - \theta(\frac{i-1}{n}, t)] ds \end{array} \right) \quad (18)$$

and

$$\mathbf{P}_{i-1}^i(t) = \begin{pmatrix} \cos \theta_M^i & \sin \theta_M^i \\ -\sin \theta_M^i & \cos \theta_M^i \end{pmatrix}. \quad (19)$$

$\theta_M^i(t) = \theta\left(\frac{i}{n}, t\right) - \theta\left(\frac{i-1}{n}, t\right)$ is the relative angle of rotation of the frame at $s = \frac{i}{n}$ with respect to the one at $s = \frac{i-1}{n}$.

Assume that the functions $\{\epsilon(s, t), \theta(s, t)\}$ have been specified. Each \mathbf{R}_{i-1}^i and \vec{r}_{i-1}^i can then be computed in parallel as a function of the backbone reference set geometry. For example, in the planar case, if we specify that $\theta(s, t) = a_1(t)s$ and $l(s, t) = a_2(t)$, then:

$$\vec{x}(s, t) = \begin{pmatrix} \frac{a_2(t)}{a_1(t)}[1 - \cos a_1(t)s] \\ \frac{a_2(t)}{a_1(t)} \sin a_1(t)s \end{pmatrix}, \quad \mathbf{Q}(s, t) = \begin{pmatrix} \cos a_1(t)s & \sin a_1(t)s \\ -\sin a_1(t)s & \cos a_1(t)s \end{pmatrix} \quad (20)$$

and

$$\vec{p}_{i-1}^i(t) = \begin{pmatrix} \frac{a_2(t)}{a_1(t)}\left[1 - \cos \frac{a_1(t)}{n}\right] \\ \frac{a_2(t)}{a_1(t)} \sin \frac{a_1(t)}{n} \end{pmatrix}, \quad \mathbf{P}_{i-1}^i(t) = \begin{pmatrix} \cos \frac{a_1(t)}{n} & \sin \frac{a_1(t)}{n} \\ -\sin \frac{a_1(t)}{n} & \cos \frac{a_1(t)}{n} \end{pmatrix}. \quad (21)$$

This provides the kinematic inputs for each module in equations (16) and (17). The inverse kinematics of each module can be performed in parallel to yield \vec{q}^{M_i} as a function of the curve geometry for each i . In this case the curve geometry is specified by values of a_1 and a_2 . Therefore, a_1 and a_2 determine manipulator configuration. An example illustrates this below.

Figure 5(a) shows one module of the planar truss manipulator. In this case, one segment of the truss is composed of side members and a cross element. The vectors representing the legs of the i th truss module defined in the frame at the center of the $(i-1)$ st face are denoted $\vec{\lambda}_{3i}, \vec{\lambda}_{3i+1}, \vec{\lambda}_{3i+2}$.

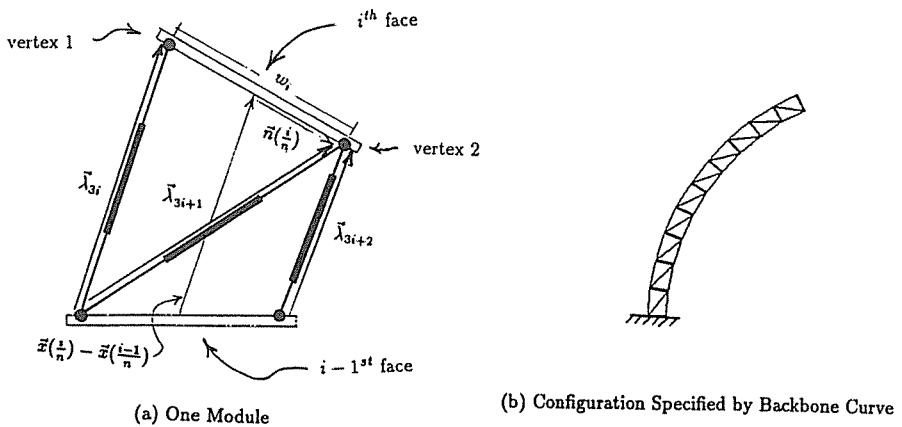


Figure 5. A planar variable geometry truss manipulator

These vectors are determined from the continuous curve model as follows:

$$\vec{\lambda}_{3i} = \vec{p}_{i-1}^i - \vec{n}_1^{i-1} + \mathbf{P}_{i-1}^i \vec{n}_1^i, \quad (22)$$

$$\vec{\lambda}_{3i+1} = \vec{p}_{i-1}^i - \vec{n}_1^{i-1} + \mathbf{P}_{i-1}^i \vec{n}_2^i, \quad (23)$$

$$\vec{\lambda}_{3i+2} = \vec{p}_{i-1}^i - \vec{n}_2^{i-1} + \mathbf{P}_{i-1}^i \vec{n}_2^i, \quad (24)$$

where \vec{n}_j^i are the vectors to the j th vertex of the i th platform in the frame affixed to that platform. For this specific example,

$$\vec{n}_1^i = [-w_i/2, 0]^T, \quad \vec{n}_2^i = [w_i/2, 0]^T,$$

where w_i is the width of each horizontal face of the truss, as denoted in Fig. 5(a).

The controlled degrees of freedom are lengths

$$\mathcal{L}_k = \|\vec{\lambda}_k\| \quad (25)$$

for $k = 1, \dots, 3n$. Thus, equations (22)–(25) provide the inverse kinematics solution for this module geometry based on the backbone curve information provided in (18)–(21). A sample configuration is shown in Fig. 5(b) for $a_1 = a_2 = 0.8$. In general, restricting the configuration of a hyper-redundant manipulator to act as if it has fewer degrees of freedom than it actually does in order to perform a task is called ‘hyper-redundancy resolution’ [4].

3. CONTINUUM FORMULATION OF HYPER-REDUNDANT MANIPULATOR DYNAMICS

The general equations of a continuum and the kinematic representation of hyper-redundant manipulator backbone curves reviewed in Section 2 are used here to formulate the approximate dynamics of hyper-redundant manipulators in efficient form. Each conservation law is addressed separately in the following subsections. Subsection 3.1 addresses the mass balance, Subsection 3.2 addresses the momentum balance and Subsection 3.3 addresses the angular momentum balance. Subsection 3.4 introduces methods for linking continuum mechanics to actual hyper-redundant manipulator dynamics, i.e. the dynamics of the continuum model is ‘projected’ onto the actual robotic structure. For the case of slender ‘snake-like’ hyper-redundant manipulators, the continuum under investigation is the backbone curve.

3.1. Inertial properties of backbone reference sets: mass balance

Approximate inertial properties can be incorporated into this model very simply. Because the description of the backbone reference set is cast within a *Lagrangian* framework, manipulator inertial properties can be approximated using models similar

to continuum mechanics. We simply define the mass density per unit curve parameter as $\rho_0(s)$. In practical terms, $\rho_0(s)$ approximately captures the inertial properties of slender ‘snake-like’ hyper-redundant manipulators. No transport of mass occurs within the manipulator. However, if the robot is actuated with hydraulics, this assumption may no longer be valid because significant amounts of fluid may flow along the manipulator.

Under the assumption that the manipulator has constant mass, the mass density per unit curve parameter, $\rho_0(s)$, will always reflect the manipulator’s macroscopic inertial properties no matter how it bends and extends. The key to understanding why this is the case is that in general $L(s, t) \neq s$. Denote the mass of a manipulator from its base to a point on the manipulator at arc length $L = L(s, t)$ to be $M(L) = M(L(s, t))$. The mass density per unit arc length is then:

$$\begin{aligned}\rho(L, t) &= \rho(L(s, t), t) = \frac{\partial M}{\partial L} = \frac{\partial M}{\partial s} \frac{\partial s}{\partial L} \\ &= \frac{\rho_0(s)}{l(s, t)} = \frac{\rho_0(s)}{1 + \epsilon(s, t)}.\end{aligned}\quad (26)$$

Thus we see that if a manipulator contracts, and $l(s, t)$ decreases, the mass density per unit arc length will increase. Likewise, when the manipulator stretches, and $l(s, t)$ increases, the mass density per unit arc length will decrease. However, the mass density per unit curve parameter s will remain constant with respect to time, and so conservation of mass is implicitly incorporated in this model. This is actually a degenerate case of equation (7), where in this case $J = 1 + \epsilon(s, t)$.

We can use this fact to transform integrals in the following way:

$$\begin{aligned}\int_{L(\sigma, t)}^{L(1, t)} \rho(L, t) \hat{f}(L, t) dL &= \int_{\sigma}^1 \rho(L(s, t), t) \hat{f}(L(s, t), t) \frac{\partial L}{\partial s} ds \\ &= \int_{\sigma}^1 \rho_0(s) f(s, t) ds\end{aligned}$$

where $\hat{f}(L(s, t), t) = f(s, t)$ and σ is an arbitrary value of the curve parameter. Therefore, all inertial terms and body forces are written in a similar form independent of whether they are expressed in terms of actual arc length, L , or referential arc length, s .

3.2. Momentum balance

The conservation of momentum equations provided by continuum mechanics take on a particular form when combined with the backbone model presented earlier. Namely:

$$\frac{d}{dt} \int_{\sigma}^1 \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds = \vec{F}(\sigma, t) + \int_{\sigma}^1 (\vec{t}_0(s, t) + \rho_0(s) \vec{b}(s, t)) ds. \quad (27)$$

The integrals over volume and surface in (3) both degenerate to one-dimensional integrals over the curve parameter. This is because surface forces and body forces are

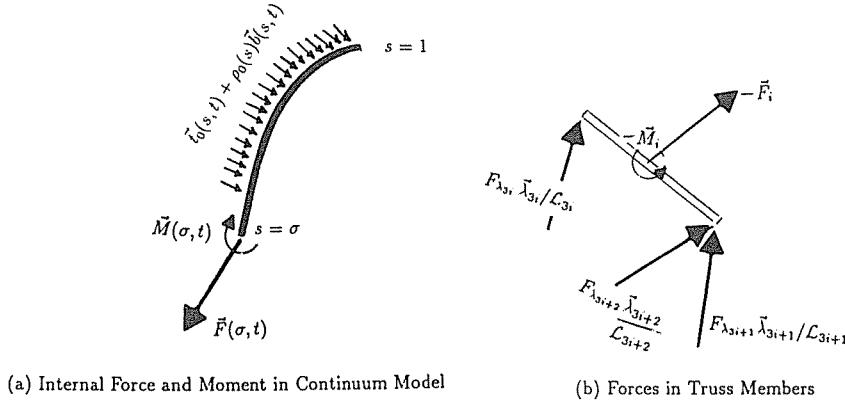


Figure 6. Free body diagrams defining ‘projection’ of dynamics

both represented as forces per unit of the backbone curve parameter. Equation (27) corresponds to a force balance on the free-body diagram in Fig. 6(a). This diagram results from an imaginary cut made normal to the backbone curve at the point at which $s = \sigma$. The vector $\vec{F}(\sigma, t)$ is the internal force transmitted to the distal end of the manipulator ($s \in [\sigma, 1]$) by the lower end of the manipulator ($s \in [\sigma, 1]$).

It should be noted that surface tractions acting on the manipulator, such as viscous drag forces, will generally be in terms of arc length, i.e. $\vec{t} = \vec{t}(L, t)$. The surface traction per unit curve parameter is derived by observing that

$$\int_{L(\sigma, t)}^{L(1, t)} \vec{t}(L, t) dL = \int_{\sigma}^1 \vec{t}(L(s, t), t) \frac{\partial L}{\partial s} ds = \int_{\sigma}^1 \vec{t}_0(s, t) ds.$$

Since this is true over all values of σ , we have that

$$\vec{t}_0(s, t) = \vec{t}(L(s, t), t) l(s, t).$$

3.3. Angular momentum balance

The angular momentum balance equation provided by continuum mechanics, equation (4), corresponding to Fig. 6(a) also has a special form for the case of hyper-redundant manipulator backbone curves:

$$\begin{aligned} \frac{d}{dt} \int_{\sigma}^1 \vec{x}(s, t) \times \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds \\ = \vec{M}(\sigma, t) + \vec{x}(\sigma, t) \times \vec{F}(\sigma, t) \\ + \int_{\sigma}^1 \vec{x}(s, t) \times (\vec{t}_0(s, t) + \rho_0(s) \vec{b}(s, t)) ds. \end{aligned} \quad (28)$$

Again referring to the imaginary cut made normal to the backbone curve at the point at which $s = \sigma$, the vector $\vec{M}(\sigma, t)$ is the internal moment transmitted to the distal end of the manipulator.

Equations (27) and (28) furnish all the tools needed to compute hyper-redundant manipulator dynamics.

3.4. Projecting dynamics onto robotic structures

In order to make use of the continuum model, there must be a way to transfer the dynamical information to the actual physical structure under consideration. In broad terms, projecting the dynamics of the continuum model onto the actual manipulator is achieved by again making an imaginary ‘cut’ in the continuum model. Only now, the forces and moments at the cut will be matched with the actual hyper-redundant structure at corresponding locations along the length of the manipulator. Inertial forces, body forces and surface tractions accumulated from the distal end of the manipulator to the cross-section under investigation will be approximated using the backbone curve model. The resulting reaction forces are calculated in the physical structure at the imaginary cutting plane. For example, the rules of structural analysis are used when considering the forces on a variable geometry truss. For manipulators with a macroscopic serial structure, the imaginary cutting planes are located at the interface between links or modules. Therefore,

$$\frac{d}{dt} \int_{i/n}^1 \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds - \int_{i/n}^1 (\vec{t}_0(s, t) + \rho_0(s) \vec{b}(s, t)) ds = \vec{F}_i(t), \quad (29)$$

$$\begin{aligned} \frac{d}{dt} \int_{i/n}^1 \vec{x}(s, t) \times \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds - \int_{i/n}^1 \vec{x}(s, t) \times (\vec{t}_0(s, t) + \rho_0(s) \vec{b}(s, t)) ds \\ - \vec{x}(i/n, t) \times \vec{F}_i(t) = \vec{M}_i(t), \end{aligned} \quad (30)$$

where \vec{F}_i and \vec{M}_i are the continuum approximation of the force and moment exerted by the i th module (or link) on the $(i+1)$ st module of a hyper-redundant manipulator.

Each of the above integrals can be evaluated separately for $i \in [0, \dots, n-1]$ and so the dynamics problem can be completely parallelized. The key to this approach is the continuum model, without which serial computations would have to be performed and a Newton–Euler style algorithm would result. With the continuum model, closed form solutions or quadrature approximations to the integrals can be computed and so there is no need for iteration.

Assuming that the inertial forces, body forces and surface tractions computed from the continuum model are representative of the actual manipulator, the reactions present in the manipulator structure at the i th module are equated to the above quantities. It is then simply a matter of matching forces in the actual structure to those generated from the continuum model, as shown in Fig. 6(b). The resulting forces in the members are found by inverting the matrix equation:

$$\begin{pmatrix} \vec{z}_{3i} \\ -[\vec{e}_3, \vec{n}(\frac{i}{n}, t), \vec{z}_{3i}] \end{pmatrix} \begin{bmatrix} \vec{z}_{3i+1} \\ [\vec{e}_3, \vec{n}(\frac{i}{n}, t), \vec{z}_{3i+1}] \end{bmatrix} \begin{bmatrix} \vec{z}_{3i+2} \\ [\vec{e}_3, \vec{n}(\frac{i}{n}, t), \vec{z}_{3i+2}] \end{bmatrix} \begin{pmatrix} F_{\lambda_{3i}} \\ F_{\lambda_{3i+1}} \\ F_{\lambda_{3i+2}} \end{pmatrix} = \begin{pmatrix} \vec{F}_i \\ \vec{e}_3, \vec{M}_i \end{pmatrix} \quad (31)$$

to solve for F_{λ_j} , which is the force in the j th member of the truss. In the above equation, the following notation is used: $[\vec{a}, \vec{b}, \vec{c}] = \vec{a} \cdot (\vec{b} \times \vec{c})$. For this particular example, F_{λ_j} are the generalized joint torques, i.e. τ_j . This information can be used in a computed torque control scheme as shown in Fig. 7, where the vectors \vec{n}_j^i are defined in Subsection 2.3.2.

Note that $\vec{n}(i/n, t) = \mathbf{Q}(i/n, t)\vec{n}_2^i$. The vectors \vec{z}_i are the unit vectors along the truss elements written in base frame coordinates. That is,

$$\vec{z}_i = -\mathbf{Q}\left(\frac{i-1}{n}, t\right) \frac{\vec{\lambda}_i}{\mathcal{L}_i}.$$

These are written explicitly as:

$$\vec{z}_{3i} = \frac{-\vec{n}\left(\frac{i-1}{n}\right) - \vec{x}\left(\frac{i}{n}\right) + \vec{x}\left(\frac{i-1}{n}\right) + \vec{n}\left(\frac{i}{n}\right)}{\|\vec{n}\left(\frac{i-1}{n}\right) + \vec{x}\left(\frac{i}{n}\right) - \vec{x}\left(\frac{i-1}{n}\right) - \vec{n}\left(\frac{i}{n}\right)\|}, \quad (32)$$

$$\vec{z}_{3i+1} = \frac{-\vec{n}\left(\frac{i-1}{n}\right) - \vec{x}\left(\frac{i}{n}\right) + \vec{x}\left(\frac{i-1}{n}\right) - \vec{n}\left(\frac{i}{n}\right)}{\|\vec{n}\left(\frac{i-1}{n}\right) + \vec{x}\left(\frac{i}{n}\right) - \vec{x}\left(\frac{i-1}{n}\right) + \vec{n}\left(\frac{i}{n}\right)\|}, \quad (33)$$

$$\vec{z}_{3i+2} = \frac{\vec{n}\left(\frac{i-1}{n}\right) - \vec{x}\left(\frac{i}{n}\right) + \vec{x}\left(\frac{i-1}{n}\right) - \vec{n}\left(\frac{i}{n}\right)}{\|-\vec{n}\left(\frac{i-1}{n}\right) + \vec{x}\left(\frac{i}{n}\right) - \vec{x}\left(\frac{i-1}{n}\right) + \vec{n}\left(\frac{i}{n}\right)\|}. \quad (34)$$

Because truss structures absorb the vast majority of the load axially in its members, there are no significant bending moments in these members.

Section 4 illustrates the general formulation of this section with a closed form example. This example is examined in the context of (i) pick-and-place tasks and (ii) inspection in hazardous sludge.

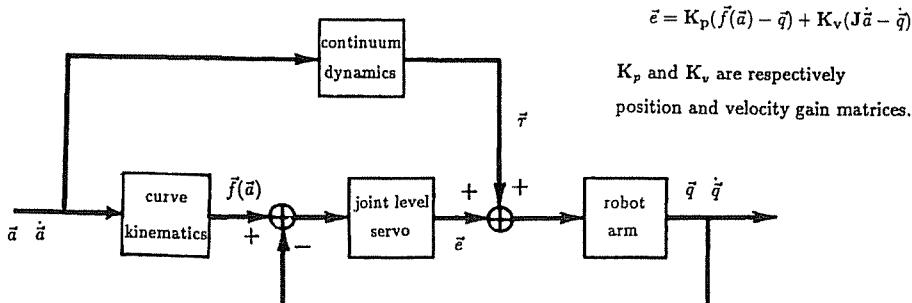


Figure 7. Control scheme using continuum model

4. EXAMPLES ILLUSTRATING APPLICATIONS

In this section, the theoretical developments presented previously are applied to two practical situations in which hyper-redundant manipulators could be used. In both examples, it is assumed that the problem is planar and that the manipulator is constrained to behave as if it has 2 d.o.f. by the algorithmic restrictions:

$$\theta(s) = a_1(t)\phi(s), \quad l(s) = a_2(t)\phi'(s). \quad (35)$$

The ‘prime’ notation denotes differentiation with respect to s . $\phi(s)$ is a strictly increasing function ($\phi'(s) > 0$ for all $s \in [0, 1]$) with $\phi(0) = 0$ and $\phi(1) = 1$. The ‘forward kinematics’ for the backbone curve representing this class of hyper-redundant manipulator configurations is:

$$x_{ee} = x_1(1, t) = \int_0^1 a_2\phi'(s) \sin(a_1\phi(s)) \, ds = \frac{a_2}{a_1}(1 - \cos a_1), \quad (36)$$

$$y_{ee} = x_2(1, t) = \int_0^1 a_2\phi'(s) \cos(a_1\phi(s)) \, ds = \frac{a_2}{a_1} \sin a_1, \quad (37)$$

with the position to points along the backbone given by:

$$x_1(s, t) = \frac{a_2}{a_1}[1 - \cos a_1\phi(s)], \quad (38)$$

$$x_2(s, t) = \frac{a_2}{a_1} \sin a_1\phi(s). \quad (39)$$

The inverse kinematics (solution for a_1 and a_2 as a function of end-effector position) is:

$$a_1 = 2\text{Atan}2(x_{ee}, y_{ee}), \quad (40)$$

and

$$a_2 = \frac{a_1 y_{ee}}{\sin a_1}, \quad (41)$$

where $\text{Atan}2(\cdot, \cdot)$ is the two argument tangent function commonly used in robotics [9]. The functions $a_1(t)$ and $a_2(t)$ are thus calculated using (40) and (41) to cause the manipulator’s end-effector to traverse a desired trajectory $(x_{ee}(t), y_{ee}(t))$. The manipulator inverse dynamics becomes a function of the two variables a_1 and a_2 and their time derivatives when algorithmic constraints such as (35) are imposed. For instance, if we take $\phi(s) = s$ and $\rho_0(s) = \rho_0$ is constant:

$$\frac{d}{dt} \int_{i/n}^1 \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) \, ds = \rho_0 \frac{d^2}{dt^2} \left(\begin{array}{l} \frac{a_2}{a_1} \left[s + \frac{\sin a_1 s}{a_1} \right]_{s=i/n}^1 \\ -\frac{a_2}{a_1} \left[\frac{\cos a_1 s}{a_1} \right]_{s=i/n}^1 \end{array} \right), \quad (42)$$

and

$$\begin{aligned} & \frac{d}{dt} \int_{i/n}^1 \vec{x}(s, t) \times \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds \\ &= \rho_0 \frac{d}{dt} \left(\frac{a_2^2}{a_1^2} \dot{a}_1 \left[\frac{1}{2} s^2 + \frac{s \sin a_1 s}{a_1} + \frac{\cos a_1 s}{a_1^2} \right]_{s=i/n}^1 \right) \vec{e}_3. \end{aligned} \quad (43)$$

4.1. Industrial pick-and-place tasks

One application of hyper-redundant manipulators is in industrial processes where objects such as tools, metal components, etc., must be picked up from one location and placed in another. In this section, the dynamics involved in performing such a process is formulated, based on the general formulation of Section 3.

In this case, it will be assumed that there are no external surface tractions acting on the manipulator. That is, the only external forces acting on the hyper-redundant manipulator are body forces (in particular gravity). The force and moment vectors acting on the distal $n - i$ modules of a hyper-redundant manipulator due to gravity will be of the form:

$$\int_{i/n}^1 \rho_0(s) \vec{g} ds = \left(1 - \frac{i}{n} \right) \rho_0 \vec{g}, \quad (44)$$

$$\begin{aligned} & \int_{i/n}^1 \vec{x}(s, t) \times (\rho_0(s) \vec{g}) ds \\ &= \rho_0 \left(g_2 \frac{a_2}{a_1} \left[s - \frac{\sin a_1 s}{a_1} \right]_{i/n}^1 + g_1 \frac{a_2}{a_1} \left[\frac{\cos a_1 s}{a_1} \right]_{i/n}^1 \right) \vec{e}_3, \end{aligned} \quad (45)$$

where \vec{g} is the vector of gravitational acceleration, and equations (38) and (39) have been used to yield a closed form solution for $\vec{x}(s, t)$ with $\phi(s) = s$. The total forces (inertial and body) which must be compensated by forces in the members of the i th bay of the truss are

$$\vec{F}_i(t) = - \int_{i/n}^1 \rho_0(s) \vec{g} ds + \frac{d}{dt} \int_{i/n}^1 \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds, \quad (46)$$

$$\begin{aligned} \vec{M}_i(t) &= - \int_{i/n}^1 \vec{x}(s, t) \times (\rho_0(s) \vec{g}) ds \\ &+ \frac{d}{dt} \int_{i/n}^1 \vec{x}(s, t) \times \rho_0(s) \frac{\partial \vec{x}}{\partial t}(s, t) ds - \vec{x}(i/n, t) \times \vec{F}_i(t), \end{aligned} \quad (47)$$

where each of the above integrals has been computed in closed form in (42)–(45), and $\vec{x}(i/n, t)$ is given by (38) and (39) for $s = i/n$. \vec{F}_i and \vec{M}_i are then used in (31) to compute forces in the truss.

4.2. Hazardous waste inspection

In some environments in which hyper-redundant manipulators will be expected to perform, viscous frictional forces will present a major contribution to what the manipulator must overcome. This subsection considers the following problem: given a hyper-redundant manipulator used for inspection in an environment filled with very viscous fluid, approximate the forces on the manipulator as it traverses through the ‘sludge’ at very slow speeds.

For this problem, a very simplified model of the fluid behavior is assumed. In particular, it is assumed that the *Reynold's number* is zero (see [16] for a detailed explanation). In essence, this means that the viscous forces are orders of magnitude greater than the fluid's inertial forces, such as would be the case of a manipulator moving very slowly in a tar-pit. In this case, surface tractions (drag forces) dominate the dynamics of the manipulator. This is in contrast to the example in the previous subsection, where gravity and inertial terms dominate.

The drag on the manipulator *per unit arc length* as it moves through the sludge can then be approximated using the slender cylinder model as:

$$t_2(L(s, t), t) = -\mu C_2 \vec{v}(s, t) \cdot \vec{u}_2(s, t) \quad (48)$$

and

$$t_1(L(s, t), t) = -\mu C_1 \vec{v}(s, t) \cdot \vec{u}_1(s, t), \quad (49)$$

where the subscript 2 represents the tangential component of drag force acting on the manipulator and 1 is the normal component. This is consistent with the convention in Section 2.3 for denoting backbone reference frame vectors. t_2 and t_1 are the components of drag in these respective directions. $\vec{v}(s, t)$ is the velocity of the backbone curve at the point with curve parameter equal to s , i.e.

$$\vec{v}(s, t) = \frac{\partial \vec{x}}{\partial t}(s, t). \quad (50)$$

μ is the viscosity of the fluid. The constants C_2 and C_1 are shape factors which have been determined in the fluid mechanics literature. It has been shown [16] that for very long cylindrical bodies,

$$C_2/C_1 \approx \frac{1}{2}. \quad (51)$$

It makes intuitive sense that pulling a cylinder through a viscous fluid lengthwise would require less effort than pushing it through sideways. We will assume that this is an approximation to a backbone curve which is ‘locally’ cylindrical.

The viscous forces *per unit curve parameter* which result in surface tractions on the manipulator are written in the base frame coordinates as:

$$\vec{t}_0(s, t) = \vec{t}(L(s, t), t)l(s, t) = (t_2(s, t)\vec{u}_2(s, t) + t_1(s, t)\vec{u}_1(s, t))l(s, t), \quad (52)$$

where in this case $l(s, t) = a_2(t)$,

$$t_2(s, t)\vec{u}_2(s, t) = -\mu C_2 \left(\frac{a_2}{a_1} \dot{a}_1 \phi(s) + \frac{d}{dt} \left(\frac{a_2}{a_1} \right) \sin a_1 \phi(s) \right) \begin{pmatrix} \sin a_1 \phi(s) \\ \cos a_1 \phi(s) \end{pmatrix} \quad (53)$$

and

$$t_1(s, t)\vec{u}_1(s, t) = -\mu C_1 \frac{d}{dt} \left(\frac{a_2}{a_1} \right) (\cos a_1 \phi(s) - 1) \begin{pmatrix} \cos a_1 \phi(s) \\ -\sin a_1 \phi(s) \end{pmatrix}, \quad (54)$$

when the manipulator shape is given by (38) and (39), where $\phi(s) = s$.

It is assumed that the manipulator is neutrally buoyant in the sludge and the surface traction due to viscosity is the only external force. In fact, if the manipulator is moving very slowly (as must be the case for (48) and (49) to hold), the inertial forces of the manipulator become insignificant, and the resulting force and moment on the distal end of the manipulator is approximated simply as:

$$\vec{F}_i(t) = - \int_{i/n}^1 \vec{t}_0(s, t) ds = \vec{F}_i^2(t) + \vec{F}_i^1(t), \quad (55)$$

$$\begin{aligned} \vec{M}_i(t) &= - \int_{i/n}^1 \vec{x}(s, t) \times \vec{t}_0(s, t) ds - \vec{x}(i/n, t) \times \vec{F}_i(t) \\ &= M_i \vec{e}_3 - \vec{x}(i/n, t) \times \vec{F}_i(t), \end{aligned} \quad (56)$$

where

$$\vec{F}_i^2 = \mu C_2 \left(\begin{array}{l} \left(\frac{a_2^2}{a_1} \dot{a}_1 \left[\frac{1}{a_1^2} \sin a_1 s - \frac{s}{a_1} \cos a_1 s \right] \right)_{s=i/n}^1 \\ \quad + \frac{d}{dt} \left(\frac{a_2}{a_1} \right) a_2 \left[\frac{s}{2} - \frac{\sin 2a_1 s}{4a_1} \right]_{s=i/n}^1 \\ \left(\frac{a_2^2}{a_1} \dot{a}_1 \left[\frac{1}{a_1^2} \cos a_1 s + \frac{s}{a_1} \sin a_1 s \right] \right)_{s=i/n}^1 \\ \quad - \frac{d}{dt} \left(\frac{a_2}{a_1} \right) a_2 \left[\frac{\cos 2a_1 s}{4a_1} \right]_{s=i/n}^1 \end{array} \right), \quad (57)$$

$$\vec{F}_i^1 = \mu C_1 \frac{d}{dt} \left(\frac{a_2}{a_1} \right) \begin{pmatrix} a_2 \left[\frac{s}{2} + \frac{\sin 2a_1 s}{4a_1} \right]_{s=i/n}^1 - y_{ee} + x_2(i/n, t) \\ a_2 \left[\frac{\cos 2a_1 s}{4a_1} \right]_{s=i/n}^1 + x_{ee} - x_1(i/n, t) \end{pmatrix}, \quad (58)$$

and

$$M_i = \frac{a_2}{a_1} \vec{F}_i \cdot \vec{e}_2 - \mu C_2 \frac{a_2^2}{a_1^2} \left[\frac{a_2}{a_1} \dot{a}_1 (1 - i^2/n^2) - \frac{d}{dt} \left(\frac{a_2}{a_1} \right) \cos a_1 s \right]_{s=i/n}^1. \quad (59)$$

In this way, the forces in the members of a variable geometry truss are again computed with (31), only now (55) and (56) are used to define \vec{F}_i and M_i , and the inertial terms are assumed to be small enough to ignore.

Thus we see that the continuum model has not only application to the dynamics of hyper-redundant manipulators in industrial environments, but also to non-standard niche applications in unstructured environments.

5. CONTINUUM VERSUS STANDARD DYNAMICS FORMULATIONS

By restricting a hyper-redundant manipulator to behave as if it possesses fewer degrees of freedom than it actually does while performing a specific task, the continuum formulation of hyper-redundant manipulator dynamics has been shown in this paper to generate simple closed-form solutions to the inverse dynamics problem. These algorithmic restrictions on a hyper-redundant manipulator's degrees of freedom are written symbolically in the form:

$$\vec{q} = \vec{f}(\vec{a}) \quad (60)$$

where $\vec{q} \in \mathbb{R}^N$ is the vector of generalized joint displacements and $\vec{a} \in \mathbb{R}^M$ is the vector of a set of weightings (such as 'modal participation factors' [12]) which specify the temporal behaviour of the backbone reference set. The function $\vec{f}(\cdot)$ contains information on the shape functions (or 'modes') chosen, kinematics of the particular hyper-redundant manipulator under consideration and the fitting procedure used. Because $M < N$, the fitting procedure (which algorithmically links the hyper-redundant manipulator to the backbone reference set) effectively restricts the behavior of the manipulator to fewer degrees of freedom than it physically possesses.

In order to achieve the same results using classical formulations of manipulator dynamics, the following steps must be followed: Differentiating (60) with respect to time, one finds:

$$\dot{\vec{q}} = \mathbf{J}(\vec{a}) \dot{\vec{a}}, \quad (61)$$

where $\mathbf{J}(\vec{a})$ is a Jacobian which relates rates of change of the modal participation factors, \vec{a} , to the generalized joint rates. Differentiating again,

$$\ddot{\vec{q}} = \mathbf{J}(\vec{a}) \ddot{\vec{a}} + \dot{\mathbf{J}}(\vec{a}) \dot{\vec{a}}. \quad (62)$$

Thus, given the mapping $\vec{f}(\cdot)$ (which is enforced via the fitting procedure), and the manipulator's dynamical equations (in the form of equation (1)), we can find the resulting inverse dynamics:

$$\mathbf{M}(\vec{f}(\vec{a}))[\mathbf{J}(\vec{a})\ddot{\vec{a}} + \dot{\mathbf{J}}(\vec{a})\dot{\vec{a}}] + \vec{C}(\vec{f}(\vec{a}), \mathbf{J}(\vec{a})\dot{\vec{a}}) + \vec{G}(\vec{f}(\vec{a})) = \vec{\tau}. \quad (63)$$

This is what the classical Lagrangian formulation of manipulator dynamics looks like for this case. The iterative Newton–Euler formulation for a hyper-redundant manipulator constrained to a backbone curve can be formulated similarly. Thus, there is a baseline for comparison between the continuum formulation and known models for computing inverse dynamics.

The following subsections formulate (63) for a lumped mass model of the variable geometry truss discussed earlier in this paper. The forces computed in the truss actuators are then compared to those generated in the continuum model.

5.1. VGT dynamics

Consider the truss shown in Fig. 5. It is assumed that the mass of this truss is concentrated at each vertex of the truss. These masses are denoted m_{ij} , where $i \in [1, \dots, n]$ denotes the module and $j \in [1, 2]$ denotes the left or right side of the truss.

The equations of motion are formulated here using Lagrange's equations. In the case of the pick-and-place task formulated using the continuum model in Subsection 4.1, the kinetic and potential energies are, respectively:

$$T = \frac{1}{2} \sum_{j=1}^2 \sum_{i=1}^n m_{ij} \dot{\vec{X}}_j^i \cdot \dot{\vec{X}}_j^i$$

and

$$V = \vec{g} \cdot \sum_{j=1}^2 \sum_{i=1}^n m_{ij} \vec{X}_j^i,$$

where \vec{X}_j^i is the position vector to the mass m_{ij} with respect to the base frame of the manipulator. The corresponding equations of motion are:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathcal{L}}_k} \right) - \frac{\partial L}{\partial \mathcal{L}_k} = F_{\lambda_k},$$

where $L = T - V$ and F_{λ_k} is the force in the k th actuatable member of the truss.

These equations are written explicitly as:

$$\sum_{j=1}^2 \sum_{i=1}^n m_{ij} \left(\ddot{\vec{X}}_j^i - \vec{g} \right) \cdot \frac{\partial \vec{X}_j^i}{\partial \mathcal{L}_k} = F_{\lambda_k},$$

where the chain rule can be used to generate:

$$\ddot{\vec{X}}_j^i = \sum_{k=1}^n \frac{\partial \vec{X}_j^i}{\partial \mathcal{L}_k} \ddot{\mathcal{L}}_k + \sum_{k=1}^n \sum_{l=1}^n \frac{\partial^2 \vec{X}_j^i}{\partial \mathcal{L}_k \partial \mathcal{L}_l} \dot{\mathcal{L}}_k \dot{\mathcal{L}}_l.$$

The inverse kinematics problem solved in equations (22)–(25) is then used to generate each leg length as a function of a_1 and a_2 . However, the vectors \vec{X}_j^i and their derivatives need to be computed via the truss forward kinematics. This is performed in the following subsection.

5.2. VGT forward kinematics

As exemplified earlier in this paper (and dealt with in detail in the author's previous work [12]) the inverse kinematics of variable geometry truss manipulators is easily solved. This, coupled with previously developed kinematic and motion planning algorithms [4, 17], allows for fast parallelizable solution of hyper-redundant manipulator inverse kinematics.

However, in order to compare the approach to hyper-redundant manipulator dynamics introduced in this paper with standard techniques, we also need to compute the forward kinematics of the variable geometry truss in order to have explicit representations of the vectors \vec{X}_j^i .

As has been documented in numerous works, the forward kinematics problem for parallel manipulators is generally much more difficult than the inverse kinematics problem. This is the reverse of the serial manipulator case in which the inverse kinematics is more complicated than the forward kinematics. For manipulators such as the variable geometry truss, which is a cascade of parallel modules, the complexity of the forward kinematics problem is a hybrid of the parallel and serial cases.

Figure 5(a) shows one module of a variable geometry truss manipulator. The forward kinematics problem for each module is the determination of the function $\mathbb{H}_{i-1}^i(\mathcal{L}_{3i}, \mathcal{L}_{3i+1}, \mathcal{L}_{3i+2})$, which maps the truss leg lengths to the position and orientation of the end-effector relative to the base frame. This can be calculated using trigonometric and/or geometric constructions. In the past, trigonometric arguments were used. Here, a simple purely graphical method will be used for the case when the truss width is the same for each module, i.e. $w_i = w$.

Consider the legs with lengths \mathcal{L}_j for $j \in [3i, 3i+1, 3i+2]$ in Fig. 5(b). Our goal is to find the positions of 'vertex 1' and 'vertex 2' as a function of leg lengths. The relative position of these vertices with respect to a frame fixed to the left corner of the base of the i th module are denoted $(x_{1,i}, y_{1,i})$ and $(x_{2,i}, y_{2,i})$, respectively. Finding the position and orientation of the top plate with respect to the bottom follow trivially once we have this information.

The following constraints apply:

$$\begin{aligned} x_{2,i}^2 + y_{2,i}^2 &= \mathcal{L}_{3i+1}^2, & (x_{2,i} - w)^2 + y_{2,i}^2 &= \mathcal{L}_{3i+2}^2, \\ x_{1,i}^2 + y_{1,i}^2 &= \mathcal{L}_{3i}^2, & (x_{1,i} - x_{2,i})^2 + (y_{1,i} + y_{2,i})^2 &= w^2. \end{aligned}$$

These equations are solved simultaneously to yield:

$$x_{2,i} = -\frac{\mathcal{L}_{3i+2}^2 - \mathcal{L}_{3i+1}^2 - w^2}{2w},$$

$$y_{2,i} = \left(\mathcal{L}_{3i+1}^2 - \left(\frac{\mathcal{L}_{3i+2}^2 - \mathcal{L}_{3i+1}^2 - w^2}{2w} \right)^2 \right)^{\frac{1}{2}}$$

and

$$x_{1,i} = \frac{-b - (b^2 - 4ac)^{\frac{1}{2}}}{2a}, \quad y_{1,i} = (\mathcal{L}_{3i}^2 - x_{1,i}^2)^{\frac{1}{2}},$$

where

$$a = 4x_{2,i}^2 + 4y_{2,i}^2, \quad b = 4(w^2 x_{2,i} - \mathcal{L}_{3i}^2 x_{2,i} - \mathcal{L}_{3i+1}^2 x_{2,i}),$$

$$c = \mathcal{L}_{3i+1}^4 + \mathcal{L}_{3i}^4 + 2\mathcal{L}_{3i}^2 \mathcal{L}_{3i+1}^2 - 2\mathcal{L}_{3i}^2 w^2 - 2\mathcal{L}_{3i+1}^2 w^2 + w^4 - 4y_{2,i}^2 \mathcal{L}_{3i}^2.$$

The forward kinematics of the whole truss structure is then expressed in terms of a sum of contributions from each truss module. The angle of inclination of the i th face with respect to the $(i-1)$ st face is given by

$$\cos(\Delta\theta_i) = \frac{x_{2,i} - x_{1,i}}{w}, \quad \sin(\Delta\theta_i) = \frac{y_{2,i} - y_{1,i}}{w},$$

and so:

$$\Delta\theta_i = \text{Atan} 2(y_{2,i} - y_{1,i}, x_{2,i} - x_{1,i}).$$

In order to represent the absolute position of each vertex of the truss in base frame coordinates, the absolute angular displacement of the i th plate is defined as:

$$\theta_i = \sum_{j=0}^i \Delta\theta_j,$$

where by definition $\Delta\theta_0 = 0$ for a fixed base. The absolute position of each vertex with respect to the center of the base of the truss is then written as:

$$\vec{X}_j^k = \vec{X}_j^{k-1} + \text{ROT}[\vec{e}_3, \theta_{k-1}] \left[\vec{x}_{k,j} + (1 + (-1)^j) \frac{w}{2} \vec{e}_1 \right].$$

$\text{ROT}[\vec{e}_3, \alpha]$ is the rotation matrix which rotates vectors counterclockwise about the \vec{e}_3 axis by an amount α and this recursive definition starts at $\vec{X}_j^0 = (-1)^j (w/2) \vec{e}_1$.

With this kinematic information, the Lagrangian model of manipulator dynamics is completed by simply taking the appropriate partial derivatives.

5.3. Numerical results

The continuum model was run together with a lumped mass model governed by Lagrange's equations. In both models, the acceleration of gravity and mass of the manipulator were set to $\vec{g} = [0, -1]^T$ and $M = 1$, respectively. It was further assumed that the mass distribution was homogeneous in both models, so in the continuum case, $\rho = 1$, and in the lumped parameter case, $m_{ij} = \frac{1}{2n}$. A test end-effector trajectory of the form:

$$x_{ee}(t) = \frac{1}{2} + \frac{1}{4} \cos(2\pi t/1000), \quad y_{ee}(t) = \frac{3}{4} + \frac{1}{4} \sin(2\pi t/1000)$$

was used for $T \in [0, 1000]$. This is a cyclic trajectory around a circle. Equations (40) and (41) convert this information into the appropriate participation factors a_1 and a_2 , which in turn specify the manipulator shape.

Results are shown in Figs 8 and 9. Fig. 8, plots of the magnitudes of the force vectors generated by the two dynamic models are compared over the trajectory. The measure used is:

$$E_1 = \frac{1}{3n} (\vec{F}_\lambda \cdot \vec{F}_\lambda)^{\frac{1}{2}} = \frac{1}{3n} |\vec{F}_\lambda|,$$

where $\vec{F}_\lambda = [F_{\lambda_0}, \dots, F_{\lambda_{3n-1}}]^T$. In this plot a truss with 10 bays was used, i.e. $n = 10$. In Fig. 9, the convergence of the two models with increasing degrees of freedom is

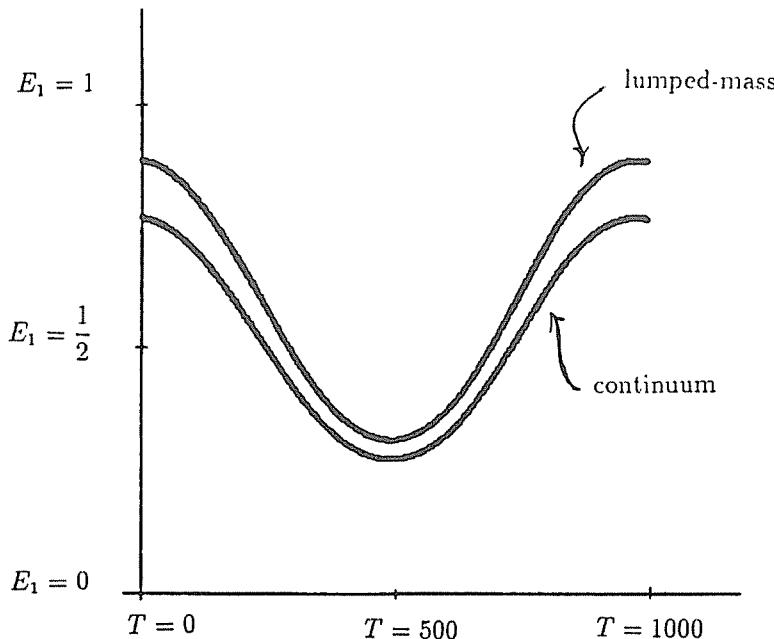


Figure 8. Comparison of the models over a circular trajectory

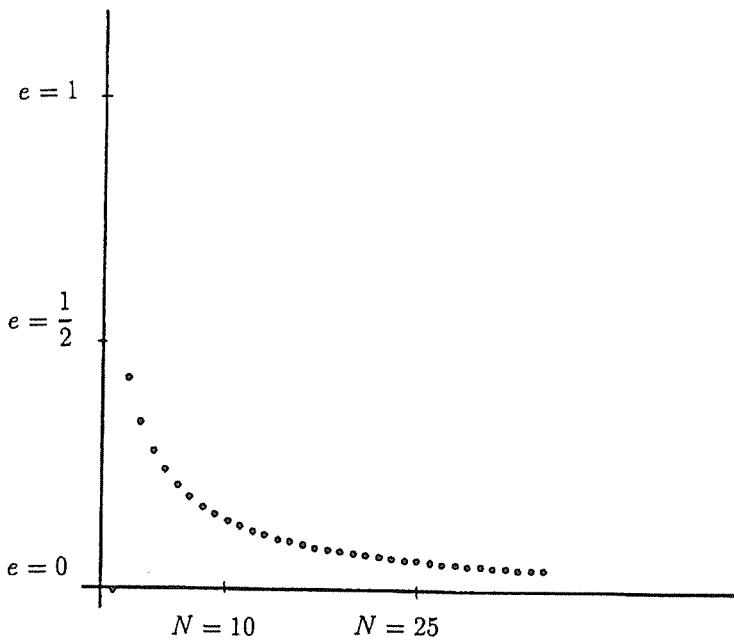


Figure 9. Convergence of the continuum and lumped-mass models

illustrated. Here the plotted quantity is

$$e = \frac{\int_0^T |\vec{F}_\lambda^l - \vec{F}_\lambda^c| dT}{\int_0^T |\vec{F}_\lambda^l| dT},$$

where the superscripts l and c denote the lumped mass and continuum models, respectively. e is the normalized difference between forces generated by the continuum and lumped mass models integrated over the trajectory.

Note the convergence of the two models between $n = 2$ and $n = 35$. Between $n = 15$ and $n = 35$, the difference between the models is less than 10%. This is extremely encouraging, because a lumped mass model is the worst case scenario to which the continuum approach can be compared. Heuristics which close this gap even further are currently being explored.

6. CONCLUSIONS

This paper has formulated the dynamics of hyper-redundant manipulators as a continuum mechanics problem. While the modeling technique is an approximation, the benefit of having expressions which can be evaluated by a highly parallel computer without any time dependence on the actual number of degrees of freedom is a powerful result. The method was demonstrated with an example of a hyper-redundant manipulator doing pick-and-place tasks in environments with gravity (such as industrial

settings) and in a viscous sludge (as may be the case in hazardous waste inspection). The accuracy of the method was verified by comparison with a Lagrangian formulation of lumped mass manipulator dynamics. It was found that the actuator forces generated in these models differed from each other by less than 10% for truss structures with between 15 and 35 bays, or 45 and 105 actuated degrees of freedom.

REFERENCES

1. G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Trans. Robotics Automat.*, in press.
2. J. W. Burdick, J. Radford and G. S. Chirikjian, "A sidewinding locomotion gait for hyper-redundant robots," *Advanced Robotics*, vol. 9, no. 3, pp. 195-216, 1995.
3. S. Hirose and Y. Umetani, "Kinematic control of active cord mechanism with tactile sensors," in *Proc. Second Int. CISM-IFT Symp. on Theory and Practice of Robots and Manipulators*, 1976, pp. 241-252.
4. G. S. Chirikjian, "Theory and applications of hyper-redundant robotic manipulators", *PhD Dissertation*, California Institute of Technology, 1992.
5. G. S. Chirikjian and J. W. Burdick, "A hyper-redundant manipulator," *IEEE Robotics Automat. Mag.*, December, pp. 22-29, 1994.
6. H. Kobayashi, E. Shimemura and K. Suzuki, "A distributed control for hyper-redundant manipulator," in *Proc. IROS'92*, Raleigh, NC, July 7-10, 1992, pp. 1958-1963.
7. D. Resnik and V. Lumelsky, "Motion planning with uncertainty for highly redundant kinematic structures I. 'Free snake' Motion," in *Proc. IROS'92*, Raleigh, NC, July 7-10, 1992, pp. 1747-1752.
8. J. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-10, no. 11, pp. 730-736, 1980.
9. J. J. Craig, *Introduction to Robotics, Mechanics and Control*. Reading, MA: Addison-Wesley, 1986.
10. H. Asada and J. J. Slotine, *Robot Analysis and Control*. New York: Wiley and Sons, 1986.
11. L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium*. Englewood Cliffs, NJ: Prentice-Hall, 1969.
12. G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robotics Automat.*, vol. 10, no. 3, pp. 343-354, 1995.
13. G. S. Chirikjian and J. W. Burdick, "Kinematically optimal hyper-redundant manipulator configurations," in *IEEE Conf. on Robotics and Automation*, Nice, France, May 1992.
14. R. S. Millman and G. D. Parker, *Elements of Differential Geometry*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
15. A. Hayashi and B. Kuipers, "A continuous approach to robot motion planning with many degrees of freedom," in *Proc. IROS'92*, Raleigh, NC, July 7-10, 1992, pp. 1935-1942.
16. R. G. Cox, "The motion of long slender bodies in a viscous fluid. Part 1: general theory," *J. Fluid Mech.*, vol. 44, pp. 791-810, 1970.
17. G. S. Chirikjian and J. W. Burdick, "Kinematics of hyper-redundant manipulators," in *Proc. ASME Mechanisms Conf.* Chicago, IL, DE-vol. 25, Sept. 16-19, 1990, pp. 391-396.
18. W. M. Lai, D. Rubin and E. Krempl, *Introduction to continuum Mechanics*. New York: Pergamon Press, 1978.
19. J. F. Wilson and U. Mahajan, "The mechanics and positioning of highly flexible manipulator limbs," *J. Mechanisms, Transmissions, Automation Des.*, vol. 111, June, 1989.

ABOUT THE AUTHOR

Gregory S. Chirikjian was born in August 1966 in New Brunswick, New Jersey, USA. He received the BSE degree in Engineering Mechanics and the MSE degree in Mechanical Engineering in 1988 while also fulfilling the requirements for a BA in Mathematics, all at The Johns Hopkins University. Between 1988 and 1992 he was a graduate student at the California Institute of Technology, where he received the PhD in Applied Mechanics in 1992. Since the summer of 1992 he has been an Assistant Professor in the Department of Mechanical Engineering at Johns Hopkins where he started the robotics program. His general research interests are in the analysis, design and implementation of 'hyper-redundant', 'metamorphic' and 'binary' manipulators. Dr Chirikjian is a National Science Foundation Young Investigator and a Presidential Faculty Fellow.

