Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 26-30, 1993

A Continuum Approach to Hyper-Redundant Manipulator Dynamics

by

Gregory S. Chirikjian
Department of Mechanical Engineering
Johns Hopkins University
Baltimore, Maryland 21218

Abstract

Hyper-redundant, or 'snakelike', 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' problem. The dynamics for this infinite dimensional case is developed. The approximate dynamics of actual hyper-redundant manipulators is then reduced to a problem which is $O(1)$ in the number of serial computations, i.e., the algorithm is $O(n)$ in the total number of computations, but these computations are completely parallelizable. 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.

1 Introduction

Hyper-redundant manipulators have a very large number of actuatable degrees of freedom. Applications of 'snakelike' hyper-redundant manipulators include inspection in highly constrained environments, tentacle-like grasping of objects, and whole-arm manipulation. Computationally attractive modeling of the system kinematics and dynamics is necessary for hyper-redundant manipulators to be used effectively. Recently, the author developed an efficient framework for the kinematics and motion planning of hyper-redundant manipulators [Ch92]. That approach is based on a continuous curve (or 'continuum') approximation which captures the manipulator's macroscopic geometric features. The continuum approach (which can be applied to a wide variety of manipulators) contrasts methods developed recently for particular hyper-redundant robot morphologies [KoSS92,ReL92].

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 [Ho80,Crai86]. 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.

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 manipulator. Section 4 illustrates this new dynamics algorithm with a closed form example.

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. Subsection 2.1 is presented primarily as a literature review, while Subsections 2.2 and 2.3 lay the analytical foundation for subsequent sections.

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:

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

The evaluation of the left-hand side of the above dynamical equations for a given trajectory in joint space, $\vec{q}(t) \in R^N$, requires $0(N^4)$ computations for a manipulator with N degrees of freedom. This is often referred to as the 'inverse dynamics' problem [AsSS6]. It has been shown that Lagrangian formulations can be improved so as to have greater computational efficiency [Ho80]. Nonetheless, the most commonly used method for formulating manipulator dynamics efficiently is the iterative Newton-Euler technique [Crai86].

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

0-7803-0823-9/93/$3.00 (C) 1993 IEEE

are given in [Crai86], and enumerated below. Outward (velocity) iterations for $i \in [0, \ldots, N-1]$ are:

$$\vec{\omega}_{i+1}^{i+1} = \mathbf{R}_i^{i+1}\vec{\omega}_i^i + \dot{q}_{i+1}\vec{z}_{i+1}^{i+1}$$

$$\dot{\vec{\omega}}_{i+1}^{i+1} = \mathbf{R}_i^{i+1}\dot{\vec{\omega}}_i^i + \mathbf{R}_i^{i+1}\vec{\omega}_i^i \times (\dot{q}_{i+1}\vec{z}_{i+1}^{i+1}) + \ddot{q}_{i+1}\vec{z}_{i+1}^{i+1}$$

$$\dot{\vec{v}}_{i+1}^{i+1} = \mathbf{R}_i^{i+1}\left(\dot{\vec{\omega}}_i^i \times \vec{r}_{i+1}^i + \vec{\omega}_i^i \times (\vec{\omega}_i^i \times \vec{r}_{i+1}^i) + \dot{\vec{v}}_i^i\right)$$

$$\dot{\vec{v}}_{C_{i+1}}^{i+1} = \dot{\vec{\omega}}_{i+1}^{i+1} \times \vec{r}_{C_{i+1}}^{i+1} + \vec{\omega}_{i+1}^{i+1} \times (\vec{\omega}_{i+1}^{i+1} \times \vec{r}_{C_{i+1}}^{i+1}) + \dot{\vec{v}}_{i+1}^{i+1}$$

$$\vec{F}_{i+1}^{i+1} = m_{i+1}\dot{\vec{v}}_{C_{i+1}}^{i+1}$$

$$\vec{N}_{i+1}^{i+1} = \mathbf{I}_{i+1}^{C}\dot{\vec{\omega}}_{i+1}^{i+1} + \vec{\omega}_{i+1}^{i+1} \times \mathbf{I}_{i+1}^{C}\vec{\omega}_{i+1}^{i+1}$$

The inward (force) iterations for $i \in [N, \ldots, 1]$ are

$$\vec{f}_i^i = \mathbf{R}_{i+1}^i\vec{f}_{i+1}^{i+1} + \vec{F}_i^i$$

$$\vec{n}_i^i = \vec{N}_i^i + \mathbf{R}_{i+1}^i\vec{n}_{i+1}^{i+1} + \vec{r}_{C_i}^i \times \vec{F}_i^i + \vec{r}_{i+1}^i \times \mathbf{R}_{i+1}^i\vec{f}_{i+1}^{i+1}$$

$$\tau_i = \vec{n}_i^i \cdot \vec{z}_i^i.$$

Most of the quantities in the iterative Newton-Euler scheme have one subscript and one superscript. The subscript indicates which link the quantity describes, while the superscript indicates in which link frame the quantity is described. For instance, $\mathbf{R}_j^k$ is the rotation matrix which describes the orientation of link $j+1$ with respect to j represented in the frame fixed to link k. $\vec{\omega}_j^k$ is the angular velocity of link j represented in the frame attached to link k. q_i is the i^{th} joint angle, i.e, $q_i = \vec{q} \cdot \vec{e}_i$. $\vec{z}_j^k$ is vector representing the axis of joint j in the frame attached to link k. $\vec{r}_j^k$ and $\vec{r}_{C_j}^k$ are respectively the relative position vectors of joint j and the mass center of link j, both written in the link k coordinate system. $\vec{v}_j^k$ is the velocity of link j. m_j and $\mathbf{I}_j^C$ are respectively the mass and moment of inertia of link j represented in link j's frame. The subscripts 'C' denote quantities associated with the center of mass of links. $\vec{F}_j^k$ and $\vec{N}_j^k$ are respectively the force and torque exerted on link j. $\vec{f}_j^k$ and $\vec{n}_j^k$ are the force and torque vectors exerted by link j on link $j+1$.

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, i.e., $\vec{\tau}$ from (1) and τ_i from above are related by the equation: $\tau_i = \vec{\tau} \cdot \vec{e}_i$ where $\vec{e}_i$ is the i^{th} natural basis vector for R^N.

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, 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

There are three general laws of continuum mechanics which will be applied to the dynamics of hyper-redundant manipulators in this paper. These are the: (1) mass balance, (2) momentum balance, and (3) angular momentum balance. These are written in control volume form respectively as:

$$\frac{d}{dt}\int_V \rho dV + \int_S \rho\vec{v} \cdot \vec{n} dS = 0 \tag{2}$$

$$\int_S \vec{t} dS + \int_V \rho\vec{b} dV = \frac{d}{dt}\int_V \rho\vec{v} dV \tag{3}$$

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

The subscripts S and V denote integrals over surface and volume of the region under consideration. ρ is the mass density per unit volume. $\vec{v}$ is the velocity of material particles. $\vec{n}$ is the normal to the control volume. $\vec{t}$ is the applied surface force (called a 'traction'). $\vec{x}$ is the position vector to material points. $\vec{b}$ is the body force acting on the volume (e.g., gravity, magnetism, etc.).

These equations have analogs in the dynamics of systems of particles. For instance, given a system of N particles, each with mass m_i, the conservation of mass states that

$$\sum_{i=1}^N m_i = M, \quad \text{or} \quad \frac{d}{dt}\sum_{i=1}^N m_i = 0,$$

where M is the total system mass.

This is the discrete analog to the first term in Equation (2). The second term in Equation (2) is the mass flux which may enter or leave the control volume under consideration. For our purposes, this term will be zero, and the continuum and discrete mass balance equations are directly analogous.

Equation (3) is the momentum balance. This is the continuous analog to the discrete equation:

$$\sum_{i=1}^N \vec{F}_i^{(e)} = \frac{d}{dt}\sum_{i=1}^N m_i\frac{d\vec{x}_i}{dt},$$

where $\vec{F}_i^{(e)}$ and $\vec{x}_i$ are respectively the external force acting on the i^{th} particle and its position vector. The right-hand term is equivalent to the product of the total system mass, M, and the acceleration of the system center of mass. The momentum balance in the continuum mechanics formulation distinguishes between forces acting on the volume and the surface, whereas in the dynamics of a system of particles there is no such distinction. Nonetheless, one can see the similarities between the discrete and continuous versions of this principle.

Finally, the angular momentum balance for a system of particles is given by:

$$\sum_{i=1}^N \vec{x}_i \times \vec{F}_i^{(e)} = \frac{d}{dt}\sum_{i=1}^N \vec{x}_i \times m_i\frac{d\vec{x}_i}{dt}.$$

This follows directly from the momentum balance for each particle by taking cross products on both sides, summing over the particles, and using the principle of action and reaction.

Note that (2)-(4) are each *postulated* separately, unlike for a system of particles where conservation of angular momentum is a direct result of conservation of linear momentum [LaRK78,Mal69]. In continuum mechanics, Equations (2)-(4) are generally transformed using the divergence and transport theorems, along with localization arguments and assumptions about applied surface forces to yield :

$$\frac{d\rho}{dt} + \rho\vec{\nabla} \cdot \vec{v} = 0 \tag{5}$$

$$\vec{\nabla} \cdot \mathbf{T} + \rho\vec{b} = \rho\frac{d\vec{v}}{dt} \tag{6}$$

$$\mathbf{T} = \mathbf{T}^{\mathbf{T}}, \tag{7}$$

where $\mathbf{T}$ is the stress tensor defined such that $\vec{t} = \mathbf{T}\vec{n}$. However, we will use the control volume form ((2)-(4)) because it will lead to the very efficient formulation of approximate hyper-redundant

dynamics which is sought. 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 Hyper-Redundant Manipulator 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 [ChB90,Ch92]. Note that depending upon 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' [ChB90], 'optimal approach' [ChB92], or any other method) can be used to directly determine the actuator displacements of a continuous morphology robot–e.g., one constructed from pneumatic actuator bundles. For discretely segmented modular morphologies, such as the one shown in Figure 1, the continuous curve solution can be used, via a 'fitting' procedure, [ChB91,Ch92], 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 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 [ChB90,Ch92].

2.3.1 Kinematics of Backbone Reference Sets

The position of points on a backbone curve can be parametrized in the form:

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

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 arclength by computing arc length in the regular way [MilP77]:

$$L(s,t) = \int_0^s [1 + \epsilon(\sigma,t)]d\sigma \tag{9}$$

Using localization arguments, it is clear that the only time s is equal to L is when $\epsilon(s,t) = 0$. 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 'growth-rate') 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

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

$$x_2(s,t) = \int_0^s l(\sigma,t)\cos\theta(\sigma,t)d\sigma \tag{11}$$

$\theta(s,t)$ is the clockwise measured angle which the tangent to the curve at point s makes with the x_2-axis at time t. Figure 2 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:

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

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

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

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

All the information contained in the planar backbone curve model is conveniently expressed as a parametrized set of 3×3 homogeneous transforms:

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

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 strictly for the kinematics of inextensible revolute-jointed kinematic chains [HaK92].

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 section details a parallelizable fitting procedure.

2.3.2 Inverse Kinematics in Parallel via 'Fitting'

A parallel algorithm developed in [ChB91], 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, H^{i-1}, to the "input," or base, of the module, and a frame, 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 H^{i-1} and 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 Figure 3). That is, equate 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 $L(s,t) \neq s$. Thus we retain control over how the modules extend, contract, and bend.

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

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

$\vec{q}^{M_i} \in \mathbf{R}^m$ 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 $i \in [1, .., n]$.

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

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

In the planar case, Equation (15) is written as:

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

and

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

where

$$\vec{p}^i_{i-1} = \begin{pmatrix} \int_{\frac{i-1}{n}}^{\frac{i}{n}} l(s,t)\sin[\theta(s,t) - \theta(\frac{i-1}{n}, t)]ds \\ \int_{\frac{i-1}{n}}^{\frac{i}{n}} l(s,t)\cos[\theta(s,t) - \theta(\frac{i-1}{n}, t)]ds \end{pmatrix} \tag{18}$$

and

$$\mathbf{P}^i_{i-1}(t) = \begin{pmatrix} \cos\theta^i_{Af} & \sin\theta^i_{Af} \\ -\sin\theta^i_{Af} & \cos\theta^i_{Af} \end{pmatrix} \tag{19}$$

$\theta^i_{Af}(t) = \theta\left(\frac{i}{n}, t\right) - \theta\left(\frac{i-1}{n}, t\right)$ 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 $\mathbf{R}^i_{i-1}$ and $\vec{r}^i_{i-1}$ 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:

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

and

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

This provides the kinematic inputs for each module in Equations (16-17). 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 4(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 $\bar{\lambda}_{3i}$, $\bar{\lambda}_{3i+1}$, $\bar{\lambda}_{3i+2}$.

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

$$\bar{\lambda}_{3i} = \vec{p}^i_{i-1} - \vec{n}^{i-1}_1 + \mathbf{P}^i_{i-1}\vec{n}^i_1 \tag{22}$$

$$\bar{\lambda}_{3i+1} = \vec{p}^i_{i-1} - \vec{n}^{i-1}_2 + \mathbf{P}^i_{i-1}\vec{n}^i_2 \tag{23}$$

$$\bar{\lambda}_{3i+2} = \vec{p}^i_{i-1} - \vec{n}^{i-1}_1 + \mathbf{P}^i_{i-1}\vec{n}^i_2, \tag{24}$$

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

$$\vec{n}^i_1 = [-w_i/2, 0]^T; \quad \vec{n}^i_2 = [w_i/2, 0]^T,$$

where w_i is the width of each horizontal face of the truss, as denoted in Figure 4(a).

The controlled degrees of freedom are the lengths

$$\mathcal{L}_k = ||\bar{\lambda}_k|| \tag{25}$$

for $k = 1, \ldots, 3n$. 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 Figure 4(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' [Ch92].

3 Continuum Formulation of Hyper-Redundant Manipulator Dynamics

The general equations of continuum mechanics 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 'snakelike' hyper-redundant manipulators, the continuum under investigation is the backbone curve.

3.1 Inertial Properties of Backbone Reference Sets : Conservation of Mass

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 solid mechanics. We simply define the mass density per unit curve parameter as $\rho(s)$. In practical terms, $\rho(s)$ approximately captures the inertial properties of slender 'snakelike' hyper-redundant manipulators. Since no transport of mass occurs within the manipulator, the flux terms in Equation (2) are zero. 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, $\rho(s)$, 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 to be

$M(L) = M(L(s,t))$. The mass density per unit arc length is then:

$$\frac{\partial M}{\partial L} = \frac{\partial M}{\partial s}\frac{\partial s}{\partial L} = \frac{\rho(s)}{l(s,t)} = \frac{\rho(s)}{1+\epsilon(s,t)} \tag{26}$$

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.

3.2 Momentum Balance

The momentum equation provided by continuum mechanics takes on a particular form when combined with the backbone model presented earlier. Namely:

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

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 both represented as forces per unit of the backbone curve parameter. Equation (27) corresponds to the free-body diagram in Figure 5(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 ($s \in [\sigma,1]$).

3.3 Angular Momentum Balance

The angular momentum equation provided by continuum mechanics (4) also has a special form for the case of hyper-redundant manipulator backbone curves:

$$\frac{d}{dt}\int_\sigma^1 \vec{x}(s,t)\times\rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds = \tag{28}$$
$$\vec{M}(\sigma,t) + \vec{x}(\sigma,t)\times\vec{F}(\sigma,t) + \int_\sigma^1 \vec{x}(s,t)\times(\vec{l}+\rho\vec{b})ds$$

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 (27-28) 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,

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

$$\frac{d}{dt}\int_{\frac{i}{n}}^1 \vec{x}(s,t)\times\rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds \tag{30}$$
$$-\int_{\frac{i}{n}}^1 \vec{x}(s,t)\times(\vec{l}+\rho\vec{b})ds - \vec{x}(i/n,t)\times\vec{F}_i = \vec{M}_i$$

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 $i \in [0,\ldots,n-1]$, 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 in many cases, 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 Figure 5(b). The resulting forces in the members are found by inverting the matrix equation :

$$\mathbf{A}_i\vec{f}_i = \vec{g}_i \tag{31}$$

where

$$\mathbf{A}_i = \begin{pmatrix} \vec{\varepsilon}_{3i} & \vec{\varepsilon}_{3i+1} & \vec{\varepsilon}_{3i+2} \\ -[\vec{e}_3,\vec{n}(i/n,t),\vec{\varepsilon}_{3i}] & [\vec{e}_3,\vec{n}(i/n,t),\vec{\varepsilon}_{3i+1}] & [\vec{e}_3,\vec{n}(i/n,t),\vec{\varepsilon}_{3i+2}] \end{pmatrix},$$

$$\vec{f}_i = \begin{pmatrix} F_{\lambda_{3i}} \\ F_{\lambda_{3i+1}} \\ F_{\lambda_{3i+2}} \end{pmatrix} \qquad \vec{g}_i = \begin{pmatrix} \vec{F}_i \\ \vec{\varepsilon}_3 \quad \vec{M}_i \end{pmatrix}$$

to solve for F_{λ_j}, which is the force in the j^{th} member of the truss. 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 Figure 6.

In Equation (31), the following notation is used: $[\vec{a},\vec{b},\vec{c}] = \vec{a}\cdot(\vec{b}\times\vec{c})$. Note that $\vec{n}(i/n,t) = \mathbf{Q}(i/n,t)\vec{n}_2^i$. The vectors $\vec{\varepsilon}_i$ are the unit vectors along the truss elements written in base frame coordinates. That is, $\vec{\varepsilon}_i = -\mathbf{Q}(\frac{i-1}{n},t)\frac{\vec{\lambda}_i}{\mathcal{L}_i}$. These are written explicitly as :

$$\vec{\varepsilon}_{3i} = \frac{-\vec{n}(\frac{i-1}{n}) - \vec{x}(\frac{i}{n}) + \vec{x}(\frac{i-1}{n}) + \vec{n}(\frac{i}{n})}{||\vec{n}(\frac{i-1}{n}) + \vec{x}(\frac{i}{n}) - \vec{x}(\frac{i-1}{n}) - \vec{n}(\frac{i}{n})||} \tag{32}$$

$$\vec{\varepsilon}_{3i+1} = \frac{\vec{n}(\frac{i-1}{n}) - \vec{x}(\frac{i}{n}) + \vec{x}(\frac{i-1}{n}) - \vec{n}(\frac{i}{n})}{||-\vec{n}(\frac{i-1}{n}) + \vec{x}(\frac{i}{n}) - \vec{x}(\frac{i-1}{n}) + \vec{n}(\frac{i}{n})||} \tag{33}$$

$$\vec{\varepsilon}_{3i+2} = \frac{-\vec{n}(\frac{i-1}{n}) - \vec{x}(\frac{i}{n}) + \vec{x}(\frac{i-1}{n}) - \vec{n}(\frac{i}{n})}{||\vec{n}(\frac{i-1}{n}) + \vec{x}(\frac{i}{n}) - \vec{x}(\frac{i-1}{n}) + \vec{n}(\frac{i}{n})||} \tag{34}$$

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

The next section illustrates the formulation of this section with a closed form example.

4 A Closed-Form Example

In this section, the theoretical developments presented previously are applied to a practical situation in which hyper-redundant manipulators could be used. It is assumed that the problem is planar, and that the manipulator is constrained to behave as if it has two degrees-of-freedom by the algorithmic restrictions :

$$\theta(s) = a_1(t)\phi(s) \quad l(s) = a_2(t)\phi'(s) \tag{35}$$

The 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:

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

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

with the position to points along the backbone given by:

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

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

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

$$a_1 = 2\Lambda\tan2(x_{ee}, y_{ee}) \tag{40}$$

$$a_2 = \frac{a_1 y_{ee}}{\sin a_1} \tag{41}$$

The functions $a_1(t)$ and $a_2(t)$ are thus calculated using (40-41) to cause the manipulator's end-effector to traverse a desired trajectory (x_{ee}, y_{ee}). 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 $\rho(s) = \rho_0$:

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

and

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

If we assume that the hyper-redundant manipulators is being used in an industrial process where objects such as tools, metal components, etc., must be picked up from one location and placed in another, then gravity forces must be considered. 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 :

$$\int_{\frac{i}{n}}^1 \rho\vec{g}ds = (1-\frac{i}{n})\rho_0\vec{g} \tag{44}$$

$$\int_{\frac{i}{n}}^1 \vec{x}(s,t)\times(\rho\vec{g})ds = \left(g_2\frac{a_2}{a_1}\left[s-\frac{\sin a_1 s}{a_1}\right]_{i/n}^1 + g_1\frac{a_2}{a_1}\left[\frac{\cos a_1 s}{a_1}\right]_{i/n}^1\right)\vec{c}_3 \tag{45}$$

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

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

$$\vec{M}_i = -\int_{\frac{i}{n}}^1 \vec{x}(s,t)\times(\rho\vec{g})ds + \frac{d}{dt}\int_{\frac{i}{n}}^1 \vec{x}(s,t)\times\rho(s)\frac{\partial\vec{x}}{\partial t}(s,t)ds \tag{47}$$
$$-\vec{x}(i/n,t)\times\vec{F}_i$$

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

It is interesting to note that this method can be viewed as the opposite of lumped mass approximations. That is, instead of component masses being lumped at discrete positions within a manipulator, the mass is 'smeared' in a continuous fashion. As shown here, this provides the opportunity to generate efficient closed form approximate solutions to complicated problems.

5 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 trajectory tracking in an environment with gravity.

6 References

[AsS86] Asada, H., Slotine, J.J., *Robot Analysis and Control*, Wiley and Sons, New York, 1986

[ChB90] Chirikjian, G.S., Burdick, J.W., "Kinematics of Hyper-Redundant Manipulators," *Proceedings of the ASME Mechanisms Conference*, Chicago, IL, DE-Vol. 25, pp. 391-396, Sept. 16-19, 1990

[ChB91] Chirikjian, G.S., Burdick, J.W., "Parallel Formulation of the Inverse Kinematics of Modular Hyper-Redundant Manipulators," *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, April, 1991.

[ChB92] Chirikjian, G.S., Burdick, J.W., " Kinematically Optimal Hyper-Redundant Manipulator Configurations," *IEEE Conference on Robotics and Automation*, Nice, France, May, 1992

[Ch92] Chirikjian, G.S., "Theory and Applications of Hyper-Redundant Robotic Manipulators," Ph.D. Dissertation, California Institute of Technology, May 1992.

[Ch93] Chirikjian, G.S., "A Continuum Model of Hyper-Redundant Manipulator Mechanics," (in preparation).

[Crai86] Craig, J.J., *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, Reading, Mass., 1986.

[HaK92] Hayashi, A., Kuipers, B., "A Continuous Approach to Robot Motion Planning with Many Degrees of Freedom," *Proceedings, IROS'92*, pp. 1935-1942, Raleigh, NC, July 7-10, 1992

[Hol80] Hollerbach, J., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-10, No. 11, pp. 730-736, Nov. 1980.

[KoSS92] Kobayashi, H., Shimemura, E., Suzuki, K., "A Distributed Control for Hyper Redundant Manipulator," *Proceedings, IROS'92*, pp. 1958-1963, Raleigh, NC, July 7-10, 1992.

[LaRK78] Lai, W.M., Rubin, D., Krempl, E., Introduction to Continuum Mechanics, Pergamon Press, New York, 1978.

[Mal69] Malvern, L.E., Introduction to the Mechanics of a Continuous Medium, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.

[MilP77] Millman, R.S., Parker, G.D., *Elements of Differential Geometry*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1977.

[ReL92] Resnik, D., Lumelsky, V., "Motion Planning with Uncertainty for Highly Redundant Kinematic Structures I 'Free Snake' Motion," *Proceedings, IROS'92*, pp. 1747-1752, Raleigh, NC, July 7-10, 1992.

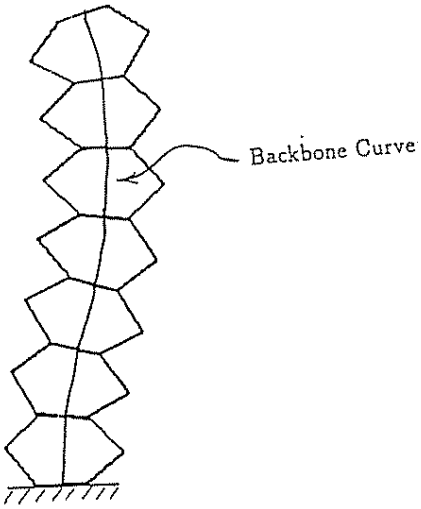


Figure 1: A Modular Morphology with Superimposed Backbone Curve

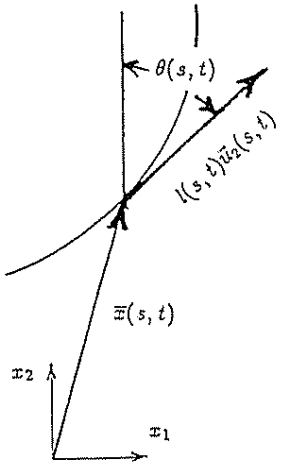


Figure 2: Description of Backbone Curve Parametrization

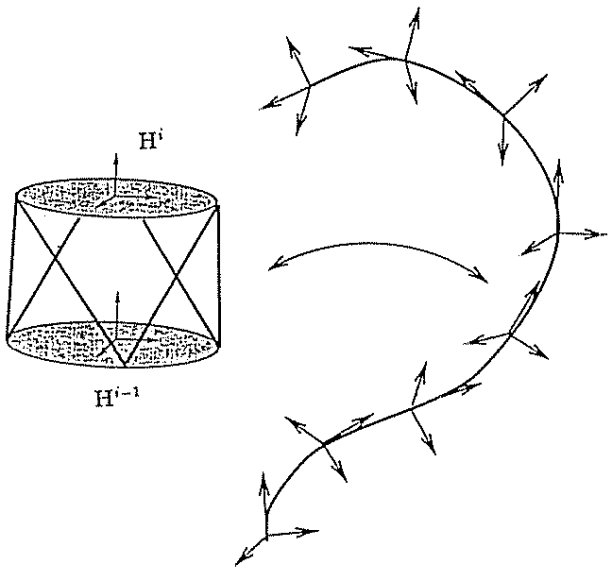
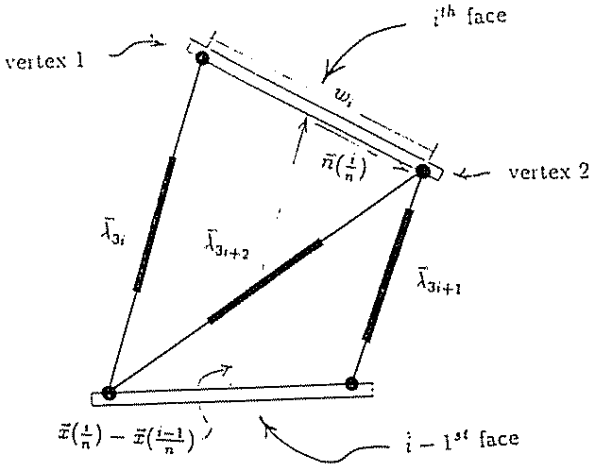
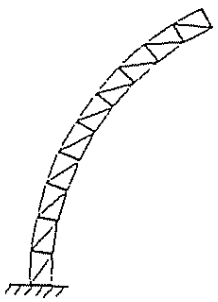


Figure 3: Fitting Manipulator Modules to the Backbone

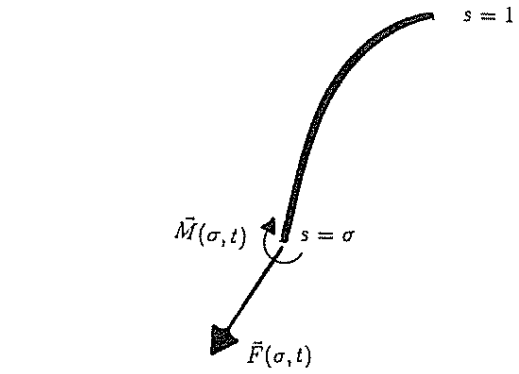


(a) One Module

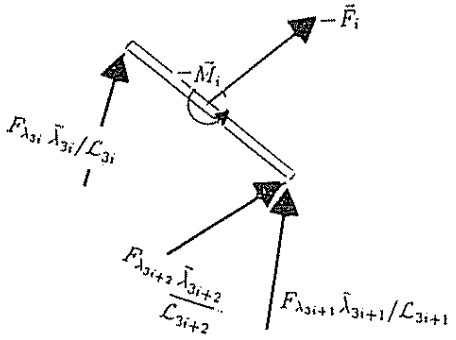


(b) Configuration Specified by Backbone Curve

Figure 4: A Planar Variable Geometry Truss Manipulator



(a) Internal Force and Moment in Continuum Model



(b) Forces in Truss Members

Figure 5: Free Body Diagrams Defining 'Projection' of Dynamics

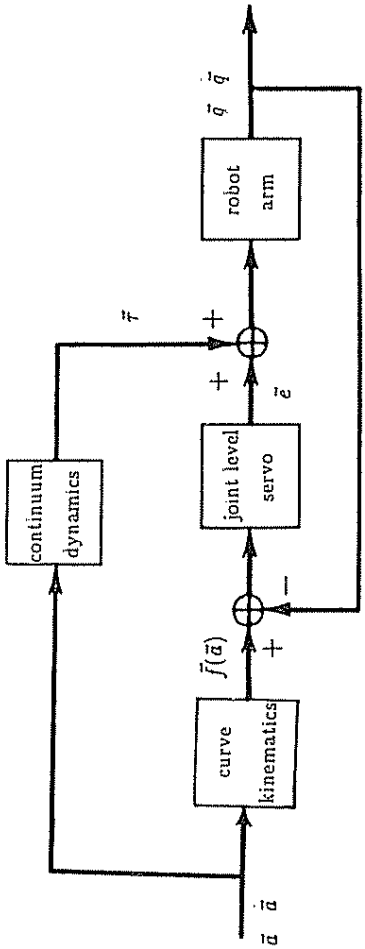


Figure 6: Control Scheme Using Continuum Model