DE-Vol. 57, Vibrations and Dynamics of Robotic
and Multibody Structures
ASME 1993

A COMPARISON OF CONTINUUM AND LUMPED-MASS MODELS FOR HYPER-REDUNDANT MANIPULATOR MECHANICS

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

Abstract

The most efficient methods for representing dynamics in the literature require serial computations which are proportional to the number of manipulator 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 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. The results are compared with a lumped mass model of a particular hyper-redundant manipulator. It is found that the continuum model can be used to generate joint torques to within ten percent of values computed from the lumped mass model.

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 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 manipulator. Section 4 illustrates this new dynamics algorithm with a closed-form example. Section 5 compares the new algorithm with a lumped-mass model

2 Background and Review

This section contains a review of a broad selection of material. First, standard techniques for formulating the dynamics of robotic manipulators are reviewed. Then we review some basic laws in continuum mechanics - an area of mechanics not commonly used in robotics. Finally, the author's previous techniques for describing hyper-redundant manipulator kinematics are reviewed.

2.1 Manipulator Dynamics and Continuum Mechanics

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:

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

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 [AsS86]. 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 can be found in any one of a number of robotics texts, e.g., [Crai86].

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.

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.). 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].

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

2.2 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 [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., such as one constructed from pneumatic actuator bundles. For discretely segmented modular morphologies 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 now reviewed. For the sake of brevity, only planar exampled are used to illustrate these concepts. For more general formulations consult [Ch92].

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

$$\vec{x}(s,t) = \int_0^s [1+\epsilon(\sigma,t)]\vec{u}(\sigma,t)d\sigma, \tag{5}$$

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 with a particular value of the curve parameter s. $\vec{u}(s,t)$ 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], which for the above parametrization is:

$$L(s,t) = \int_0^s [1+\epsilon(\sigma,t)]d\sigma. \tag{6}$$

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 (5) 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 $\vec{u}(s,t)$ and varying its magnitude (or 'growth-rate') according to $l(s,t)$.

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

$$x_1(s,t) = \int_0^s l(\sigma,t)\sin\theta(\sigma,t)d\sigma \tag{7}$$

$$x_2(s,t) = \int_0^s l(\sigma,t)\cos\theta(\sigma,t)d\sigma. \tag{8}$$

$\theta(s,t)$ is the clockwise measured angle which the tangent to the curve at point s makes with the x_2-axis at time t. Figure 1 illustrates the physical meaning of $l(s,t)$ and $\theta(s,t)$.

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 assumptions that the manipulator has constant mass and no mass transport occurs, 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:

$$\frac{\partial M}{\partial L} = \frac{\partial M}{\partial s}\frac{\partial s}{\partial L} = \frac{\rho(s)}{l(s,t)} = \frac{\rho(s)}{1+\epsilon(s,t)} \tag{9}$$

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:

$$\frac{d}{dt}\int_{\sigma}^{1} \rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds = \vec{F}(\sigma,t) + \int_{\sigma}^{1}(\vec{f}+\rho\vec{b})ds \tag{10}$$

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 (10) corresponds to the free-body diagram in Figure 2. 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:

$$\frac{d}{dt}\int_{\sigma}^{1} \vec{x}(s,t)\times\rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds = \tag{11}$$

$$\vec{M}(\sigma,t) + \vec{x}(\sigma,t)\times\vec{F}(\sigma,t) + \int_{\sigma}^{1}\vec{x}(s,t)\times(\vec{f}+\rho\vec{b})ds.$$

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 (10-11) 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 real 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,

$$\frac{d}{dt}\int_{\frac{i}{n}}^{1} \rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds - \int_{\frac{i}{n}}^{1}(\vec{f}+\rho\vec{b})ds = \vec{F}_i \tag{12}$$

$$\frac{d}{dt}\int_{\frac{i}{n}}^{1} \vec{x}(s,t)\times\rho(s)\frac{\partial \vec{x}}{\partial t}(s,t)ds \tag{13}$$

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

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. For manipulator modules like those shown in Figure 3(a), the forces are matched as shown in Figure 3(b). The resulting forces in the members are found by inverting the matrix equations :

$$\mathrm{A}_i\vec{f}_i = \vec{g}_i \tag{14}$$

for $i \in [1,\ldots,n]$, where

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

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

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.

In Equation (14), the following notation is used: $[\vec{a},\vec{b},\vec{c}] = \vec{a}\cdot(\vec{b}\times\vec{c})$. Note that $\vec{n}(i/n,t) = \frac{w}{2}[-\cos\theta(i/n,t),\sin\theta(i/n,t)]^T$, where w is the width of the truss. The vectors $\vec{z}_i$ are the unit vectors along the truss elements written in base frame coordinates. These are written explicitly as :

$$\vec{z}_{3i}(t) = \frac{-\vec{n}(\frac{i-1}{n},t) - \vec{x}(\frac{i}{n},t) + \vec{x}(\frac{i-1}{n},t) + \vec{n}(\frac{i}{n},t)}{||\vec{n}(\frac{i-1}{n},t) + \vec{x}(\frac{i}{n},t) - \vec{x}(\frac{i-1}{n},t) - \vec{n}(\frac{i}{n},t)||} \tag{15}$$

$$\vec{z}_{3i+1}(t) = \frac{\vec{n}(\frac{i-1}{n},t) - \vec{x}(\frac{i}{n},t) + \vec{x}(\frac{i-1}{n},t) - \vec{n}(\frac{i}{n},t)}{||-\vec{n}(\frac{i-1}{n},t) + \vec{x}(\frac{i}{n},t) - \vec{x}(\frac{i-1}{n},t) + \vec{n}(\frac{i}{n},t)||} \tag{16}$$

$$\vec{z}_{3i+2}(t) = \frac{-\vec{n}(\frac{i-1}{n},t) - \vec{x}(\frac{i}{n},t) + \vec{x}(\frac{i-1}{n},t) - \vec{n}(\frac{i}{n},t)}{||\vec{n}(\frac{i-1}{n},t) + \vec{x}(\frac{i}{n},t) - \vec{x}(\frac{i-1}{n},t) + \vec{n}(\frac{i}{n},t)||} \tag{17}$$

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 :

$$\theta(s,t) = a_1(t)\phi(s) \quad l(s,t) = a_2(t)\phi'(s). \tag{18}$$

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:

$$x_{ee} = x_1(1,t) = \int_0^1 a_2\phi'(s)\sin(a_1\phi(s))ds = \frac{a_2}{a_1}(1-\cos a_1) \tag{19}$$

$$y_{ee} = x_2(1,t) = \int_0^1 a_2\phi'(s)\cos(a_1\phi(s))ds = \frac{a_2}{a_1}\sin a_1, \tag{20}$$

with the position to points along the backbone given by:

$$x_1(s,t) = \frac{a_2}{a_1}\left[1 - \cos a_1\phi(s)\right] \tag{21}$$

$$x_2(s,t) = \frac{a_2}{a_1}\sin a_1\phi(s) \tag{22}$$

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

$$a_1 = 2\text{Atan2}(x_{ee}, y_{ee}) \tag{23}$$

$$a_2 = \frac{a_1 y_{ee}}{\sin a_1} \tag{24}$$

The functions $a_1(t)$ and $a_2(t)$ are thus calculated using (23-24) 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 (18) are imposed. For instance, if we take $\phi(s) = s$ and $\rho(s) = \rho_0$:

$$\frac{d}{dt}\int_{\frac{i}{n}}^1 \rho(s)\frac{\partial\vec{x}}{\partial t}(s,t)ds = \rho_0\frac{d^2}{dt^2}\begin{pmatrix} \frac{a_2}{a_1}\left[s + \frac{\sin a_1 s}{a_1}\right]_{s=i/n}^1 \\ -\frac{a_2}{a_1}\left[\frac{\cos a_1 s}{a_1}\right]_{s=i/n}^1 \end{pmatrix}, \tag{25}$$

and

$$\frac{d}{dt}\int_{\frac{i}{n}}^1 \vec{x}(s,t) \times \rho(s)\frac{\partial\vec{x}}{\partial t}(s,t)ds = \tag{26}$$

$$\rho_0\frac{d}{dt}\left(\frac{a_2^2}{a_1^2}\dot{a}_1\left[\frac{1}{2}s^2 + \frac{s\sin a_1 s}{a_1} + \frac{\cos a_1 s}{a_1^2}\right]_{s=i/n}^1\right)\vec{e}_3$$

If we assume that the hyper-redundant manipulator 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 :

$$\int_{\frac{i}{n}}^1 \rho\vec{g}ds = (1 - \frac{i}{n})\rho_0\vec{g} \tag{27}$$

$$\int_{\frac{i}{n}}^1 \vec{x}(s,t) \times (\rho\vec{g})ds = \left(g_2\frac{a_2}{a_1}\left[s - \frac{\sin a_1 s}{a_1}\right]_{i/n}^1 + g_1\frac{a_2}{a_1}\left[\frac{\cos a_1 s}{a_1}\right]_{i/n}^1\right)\vec{e}_3 \tag{28}$$

where $\vec{g}$ is the vector of gravitational acceleration, and Equations (21-22) 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

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

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

$$-\vec{x}(i/n,t) \times \vec{F}_i$$

where each of the above integrals has been computed in closed form in (25-28), and $\vec{x}(i/n,t)$ is given by (21-22) for $s = i/n$. $\vec{F}_i$ and $\vec{M}_i$ are then used in (14) 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 Continuum vs. 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:

$$\vec{q} = \vec{f}(\vec{a}) \tag{31}$$

where $\vec{q} \in R^N$ is the vector of generalized joint displacements, and $\vec{a} \in R^M$ is the vector of weightings (or 'modal participation factors') which specify the temporal behavior 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 formulate the Lagrangian dynamics problem for a hyper-redundant manipulator which is algorithmically constrained by (31), the following steps must be followed. Differentiating (31) with respect to time, one finds :

$$\dot{\vec{q}} = \mathrm{J}(\vec{a})\dot{\vec{a}} \tag{32}$$

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

$$\ddot{\vec{q}} = \mathrm{J}(\vec{a})\ddot{\vec{a}} + \dot{\mathrm{J}}(\vec{a})\dot{\vec{a}} \tag{33}$$

Thus, given the mapping $\vec{f}(\cdot)$ (which is enforced via algorithmically linking the backbone curve and manipulator geometries), and the manipulator's dynamical equations, (in the form of Equation (1)), we can find the resulting inverse dynamics:

$$\mathrm{M}(\vec{f}(\vec{a}))[\mathrm{J}(\vec{a})\ddot{\vec{a}} + \dot{\mathrm{J}}(\vec{a})\dot{\vec{a}}] + \vec{C}(\vec{f}(\vec{a}), \mathrm{J}(\vec{a})\dot{\vec{a}}) + \vec{G}(\vec{f}(\vec{a})) = \vec{\tau} \tag{34}$$

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 (34) 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 Variable Geometry Truss Dynamics

Consider the truss shown in Figure 3. It is assumed that the mass of this truss is concentrated, or "lumped," at each vertex of the truss. These masses are denoted m_{ij}, where $i \in [1, \ldots, n]$ 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. The kinetic and potential energies are respectively:

$$T = \frac{1}{2} \sum_{j=1}^{2} \sum_{i=1}^{n} m_{ij} \dot{\vec{X}}_j^i \cdot \dot{\vec{X}}_j^i \tag{35}$$

and

$$V = \vec{g} \cdot \sum_{j=1}^{2} \sum_{i=1}^{n} m_{ij} \vec{X}_j^i . \tag{36}$$

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:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathcal{L}}_k}\right) - \frac{\partial L}{\partial \mathcal{L}_k} = F_{\lambda_k} \tag{37}$$

where $L = T - V$, F_{λ_k} is the force in the k^{th} actuatable member of the truss, and $\mathcal{L}_k$ is the length of the kth actuatable truss member.

These equations are written explicitly as :

$$\sum_{j=1}^{2} \sum_{i=1}^{n} m_{ij} \left(\ddot{\vec{X}}_j^i - \vec{g}\right) \cdot \frac{\partial \vec{X}_j^i}{\partial \mathcal{L}_k} = F_{\lambda_k} \tag{38}$$

where the chain rule can be used to generate:

$$\ddot{\vec{X}}_j^i = \sum_{k=1}^{n} \frac{\partial \vec{X}_j^i}{\partial \mathcal{L}_k} \ddot{\mathcal{L}}_k + \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial^2 \vec{X}_j^i}{\partial \mathcal{L}_k \partial \mathcal{L}_l} \dot{\mathcal{L}}_k \dot{\mathcal{L}}_l \tag{39}$$

The inverse kinematics problem for each truss module is then used to generate each leg length as a function of a_1 and a_2 [Ch92]. However, the vectors $\vec{X}_j^i$ and their derivatives need to be computed via the truss forward kinematics. This is outlined in the following subsection.

5.2 Variable Geometry Truss Forward Kinematics

As exemplified earlier in this paper (and dealt with in detail in the author's previous work [Ch92]) the inverse kinematics of variable geometry truss manipulators is easily solved. This, coupled with previously developed kinematic and motion planning algorithms [Ch92,ChB90], 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, e.g. [Ch92], 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 3(a) shows one module of a variable geometry truss manipulator. The forward kinematics problem for each module is the determination of the function $\mathrm{H}_{i-1}^{i}(\mathcal{L}_{3i}, \mathcal{L}_{3i+1}, \mathcal{L}_{3i+2})$, which is the homogeneous transform 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. Here, a simple purely graphical method will be used.

Consider the legs with lengths $\mathcal{L}_j$ for $j \in [3i, 3i+1, 3i+2]$. 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. For a more complete explaination, see [ChH93]. In that paper, the kinematics and design issues pertaining to a variable geometry truss manipulator with binary, or 'on-off,' actuation are examined.

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 (23-24) convert this information into the appropriate participation factors, which in turn specify the manipulator shape.

Results are shown in Figures 4 and 5. In Figure 4, plots of the magnitudes of the force vectors generated by the two dynamic models are compared over the trajectory. The measure used is:

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

where $\vec{F}_\lambda = [F_{\lambda_0}, \ldots, F_{\lambda_{3n-1}}]^T$. In this plot a truss with ten bays was used, i.e., $n = 10$. In Figure 5, the convergence of the two models with increasing degrees of freedom is 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 ten percent. This is extremely encouraging, because a lumped mass model is the worst case scenario to which the continuum approach can be compared. It is believed that numerical round off errors in the lumped mass model account for the divergence which begins to occur at very large numbers of degrees of freedom

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. 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 ten percent for truss structures with between 15 and 35 bays, or 45 and 105 actuated degrees of freedom.

7 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.

[ChH93] Chirikjian, G.S., Houck, J , "Kinematics and Design of a Variable Geometry Truss with Binary Actuation," (in preparation).

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

[Ho80] 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., 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., Parker, G., 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.

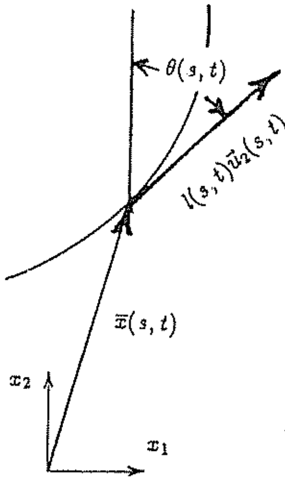


Figure 1: Description of Backbone Curve Parametrization

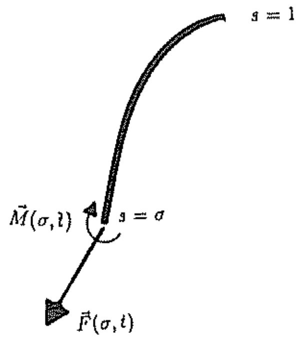
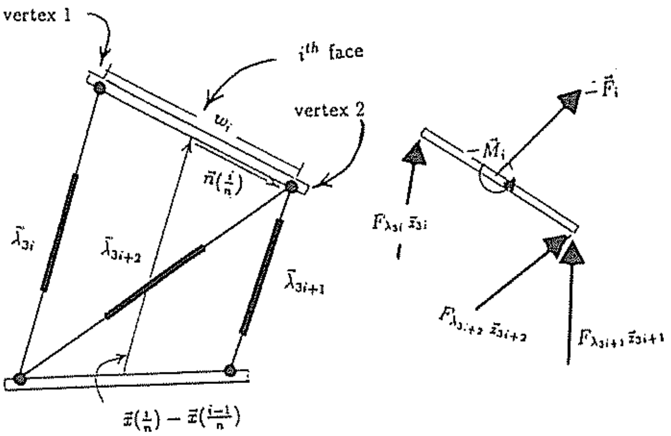


Figure 2: Internal Force and Moment in Continuum Model



vertex 1
i^{th} face
vertex 2
w_i
$\vec{n}(\frac{i}{n})$
$\vec{\lambda}_{3i}$
$\vec{\lambda}_{3i+2}$
$\vec{\lambda}_{3i+1}$
$\vec{x}(\frac{i}{n}) - \vec{x}(\frac{i-1}{n})$
$-\vec{F}_i$
$-\vec{M}_i$
$F_{\lambda_{3i}}\vec{z}_{3i}$
$F_{\lambda_{3i+2}}\vec{z}_{3i+2}$
$F_{\lambda_{3i+1}}\vec{z}_{3i+1}$

(a) One Module

(b) Forces in Truss Members

Figure 3

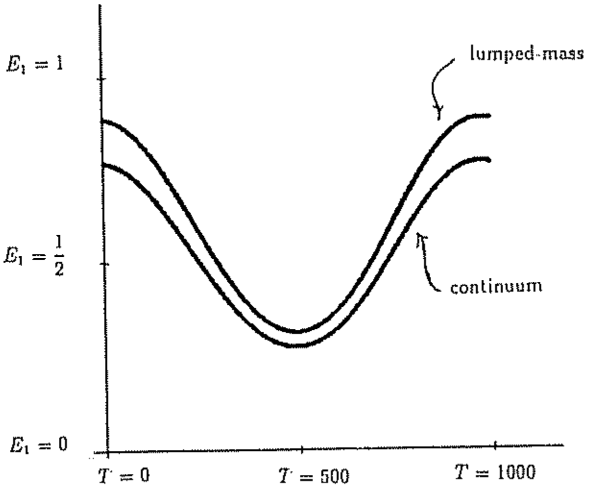


Figure 4 Comparison of the Models over a Trajectory

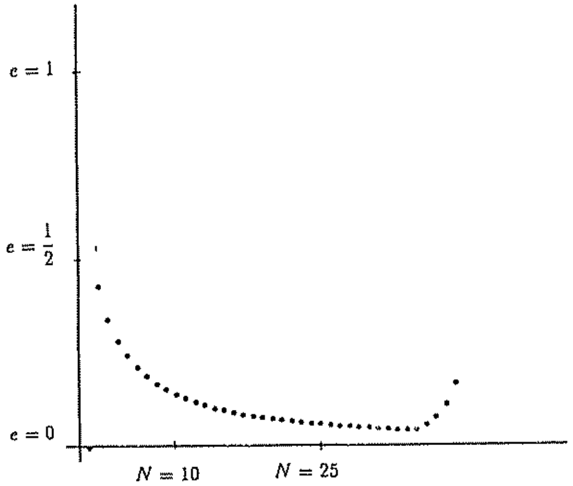


Figure 5 Convergence of the Models