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

General Methods for Computing Hyper-Redundant Manipulator Inverse Kinematics

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

ABSTRACT

"Hyper-redundant" robots have a very large or infinite degree of kinematic redundancy. This paper formulates generalized resolved rate methods for solving hyper-redundant manipulator inverse kinematics using a 'backbone curve.' These methods are applicable even in cases when explicit representation of the backbone curve intrinsic geometry cannot be written in closed form. Problems of end-effector trajectory tracking and singularity analysis which were previously intractable can now be handled easily. Examples include configurations generated using the calculus of variations. Also, the method is naturally parallelizable for fast digital and/or analog computation.

1 Introduction and Background

In recent work, a general kinematics and motion planning framework for hyper-redundant robotic manipulators has been developed [Ch92,ChB90-92]. The complexity of tasks such as end-effector placement and trajectory generation is reduced using these methods. The basis for these algorithms is straight forward : Only allow hyper-redundant manipulators to act as if they are kinematically sufficient while performing a particular task. This is referred to as hyper-redundancy resolution. General numerical techniques are used here in combination with the concept of hyper-redundancy resolution.

This paper is organized as follows: The remainder of this section reviews the parametrization of hyper-redundant manipulator 'backbone curves,' and provides motivational examples for the current work. Section 2 reviews standard numerical techniques for the solution of boundary value problems. Section 3 develops a general *intrinsic* technique for hyper-redundant manipulator inverse kinematics. Section 4 illustrates this technique with examples. Section 5 presents an alternate *extrinsic* technique for computing inverse kinematics.

1.1 Hyper-Redundant Manipulator 'Backbone Curve' Kinematics

Consider the 'infinitely redundant' planar manipulator as shown in Figure 1. The actuatable degrees of freedom of this manipulator cannot be represented with a finite length vector of joint angles, but rather consist of two functions $\theta(s,t)$ and $\epsilon(s,t)$. At each point denoted by the parameter s along the length of the manipulator, $\theta(s,t)$ controls how the manipulator bends, while $\epsilon(s,t)$ controls how the manipulator extends and contracts. The position of every point on this planar manipulator with respect to a base frame is given by $\vec{x}(s,t) = [x_1(s,t),\ x_2(s,t)]^T$, where

$$x_1(s,t) = \int_0^s [1+\epsilon(\sigma,t)]\sin\theta(\sigma,t)d\sigma \tag{1}$$

$$x_2(s,t) = \int_0^s [1+\epsilon(\sigma,t)]\cos\theta(\sigma,t)d\sigma. \tag{2}$$

$\theta(s,t)$ can be related to the classical curvature function, $\kappa(s,t)$, by observing :

$$\theta(s,t) = \int_0^s [1+\epsilon(\sigma,t)]\kappa(\sigma,t)d\sigma. \tag{3}$$

In the spatial case, four functions are needed to fully specify manipulator configuration, and

$$\vec{x}(s,t) = \begin{pmatrix} \int_0^s[1+\epsilon(\sigma,t)]\sin K(\sigma,t)\cos T(\sigma,t)d\sigma \\ \int_0^s[1+\epsilon(\sigma,t)]\cos K(\sigma,t)\cos T(\sigma,t)d\sigma \\ \int_0^s[1+\epsilon(\sigma,t)]\sin T(\sigma,t)d\sigma \end{pmatrix}. \tag{4}$$

$K(s,t)$ and $T(s,t)$ are angles which determine the direction of the tangent to the curve representing the manipulator at every point, while ϵ again specifies extensibility. By convention, the initial conditions $K(0,t) = T(0,t) = 0$ are assumed. One final function, the *roll distribution*, $R(s,t)$, is defined to specify how an actual mechanism twists about the curve $\vec{x}(s,t)$. These intrinsic functions form a vector denoted : $\vec{\theta}$. In the planar case, $\vec{\theta} = [\theta,\epsilon]^T$, while in the spatial case $\vec{\theta} = [K,T,R,\epsilon]^T$. Note that the classical arc length measure, L, is related to the curve parameter, s, through the extensibility:

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

In addition to this idealized infinite degree of freedom case, (1-3) and (4) are used to define continuous 'backbone curves' for discrete hyper-redundant manipulators with a finite number of degrees of freedom. An appropriate 'fitting' procedure is then implemented to algorithmically link the real manipulator and backbone curve kinematics [ChB91a,Ch92].

In the author's previous work, reduction of kinematic and motion planning complexity for both continuous and discrete hyper-redundant kinematic structures resulted from restrictions of the form :

$$\vec{\theta}(s,t) = \tilde{\vec{\theta}}(s,\vec{\mu}(t)) \tag{6}$$

for $\vec{\mu} \in \mathrm{R}^{\mathrm{N}}$. N is the number of end-effector coordinates. For positioning in the plane $N=2$, while for position and orientation in the plane $N=3$. In space, $N=3$ for positioning, and $N=6$ for position and orientation.

As task requirements change, these artificial restrictions imposed on hyper-redundant manipulator configuration are allowed to change also. In [ChB90a] it was first shown how *closed form* forward and inverse kinematic algorithms based on this method can

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

be used for hyper-redundant manipulators. When closed form solutions cannot be obtained, a method analogous to rate linearized (or instantaneous) kinematics for kinematically sufficient manipulators was used with great success for both trajectory generation and analysis of algorithmic singularities [ChB90b].

However, in a broader context, there are situations when these previous methods need to be augmented. This paper considers the case when an explicit algebraic representation of the functions $\vec{\theta}(s,t)$ is not available. In some situations only systems of differential equations of the form:

$$\vec{F}(s,\vec{\mu},\vec{\theta},\frac{\partial\vec{\theta}}{\partial s},\frac{\partial^2\vec{\theta}}{\partial s^2}) = \vec{0} \tag{7}$$

with initial conditions

$$\vec{G}(0,\vec{\mu},\vec{\theta}(0,\vec{\mu}),\frac{\partial\vec{\theta}(0,\vec{\mu})}{\partial s}) = \vec{0} \tag{8}$$

are available, where $\vec{F}(\cdot) \in \mathrm{R}^{\mathrm{M}}$, $\vec{G}(\cdot) \in \mathrm{R}^{2\mathrm{M}}$, and $M = \dim(\vec{\theta})$. While the methods developed in this paper are general, all the examples are for the planar case.

1.2 Examples

In the author's previous papers, techniques for resolving hyper-redundancy were presented. Two of these approaches are the 'modal approach' [ChB90b] and the calculus of variations optimality based approach [ChB92a].

A planar example of the modal approach for *nonextensible* manipulators, i.e., manipulators with $\epsilon(s,t) = 0$, is

$$\theta(s,t) = \sum_{i=1}^{N} \mu_i(t)\Phi_i(s). \tag{9}$$

In this way the end-effector position and orientation is a function of $\{\mu_i\}$. In the planar nonextensible case, the calculus of variations approach seeks to minimize integrals such as

$$I = \int_0^1 \kappa^2 ds = \int_0^1 \dot{\theta}^2 ds, \tag{10}$$

subject to the end-effector constraints:

$$x_{ee} = \int_0^1 \sin\theta ds \quad y_{ee} = \int_0^1 \cos\theta ds. \tag{11}$$

Differential equations such as:

$$\ddot{\theta} - \mu_1 \cos\theta + \mu_2 \sin\theta = 0 \tag{12}$$

together with integral constraints (11) and initial conditions

$$\theta(0,t) = 0 \quad \dot{\theta}(0,t) = \mu_3 \tag{13}$$

result when using the calculus of variations (see [ChB92a] for a derivation and the appendix for background material). Here, and throughout this paper, a $\dot{}$ represents $\frac{d}{ds}$.

In either (9) or (12-13), several free variables map to the end-effector position and/or orientation. The differential relationship between free parameters $\{\mu_i\}$, which will be referred to as the *reduced set*, and end-effector coordinates is :

$$\Delta\vec{x}_{ee} = \mathrm{J}(\vec{\mu})\Delta\vec{\mu}. \tag{14}$$

In the planar case, $\vec{x}_{ee}(t) = [\vec{x}^T(1,t),\theta(1,t)]^T$, and

$$\mathrm{J}(\vec{\mu}) = \begin{pmatrix} \int_0^1 \frac{\partial\theta}{\partial\mu_1}\cos\theta ds & \int_0^1 \frac{\partial\theta}{\partial\mu_2}\cos\theta ds & \int_0^1 \frac{\partial\theta}{\partial\mu_3}\cos\theta ds \\ -\int_0^1 \frac{\partial\theta}{\partial\mu_1}\sin\theta ds & -\int_0^1 \frac{\partial\theta}{\partial\mu_2}\sin\theta ds & -\int_0^1 \frac{\partial\theta}{\partial\mu_3}\sin\theta ds \\ \frac{\partial\theta}{\partial\mu_1} & \frac{\partial\theta}{\partial\mu_2} & \frac{\partial\theta}{\partial\mu_3} \end{pmatrix}. \tag{15}$$

This relationship can be inverted to solve for incremental changes in the free parameters as a function of end-effector changes, e.g.,

$$\Delta\vec{\mu} = \mathrm{J}^{-1}(\vec{\mu})\Delta\vec{x}_{ee}, \tag{16}$$

in much the same way that kinematically sufficient manipulator kinematics is dealt with. It is clear that given an explicit function

$$\theta(s,t) = \hat{\theta}(s,\vec{\mu}(t)), \tag{17}$$

(15-16) can be computed numerically. The remainder of this paper addresses the issue of how such equations can be computed when a closed form representation of (17) cannot be written in terms of standard functions, such as is the case in (12-13). But first, standard numerical techniques for solving boundary value problems are reviewed.

2 Boundary Value Problems

In this section, mathematical and numerical techniques commonly used for the numerical solution of boundary value problems are briefly reviewed. Three numerical techniques are the most common : 'shooting' (also called initial value) methods, finite difference methods, and integral equation methods. These techniques are enumerated to give a comparison of potential solution strategies to the problem stated in the previous section.

The idea behind shooting is straight forward: since it is relatively easy to solve initial value problems, guess at initial conditions which may or may not make the differential equation satisfy conditions at the far boundary. Then, iteratively correct this initial guess based on the error between the desired and actual boundary conditions.

In finite difference methods, both the domain over which the problem is defined, and derivatives in the equations are discretized. This results in a (generally large) system of algebraic equations which can be inverted (either explicitly or iteratively) to compute an approximate solution to the problem at a finite number of points. If the initial differential equations are linear, the resulting finite difference equations will be as well.

Integral equation techniques are most commonly used for the numerical solution of linear differential equations. A Green's function can be found for the particular linear operator, and an integral of the product of the Green's function, and forcing terms is approximated with a quadrature algorithm to yield an approximate numerical solution.

The method which will be used here is a form of shooting which is particularly natural in the context of manipulator end-effector trajectory tracking. For more information about shooting, and all numerical methods for solving boundary value problems, see [KuH83], [Ke68], and [Me73].

3 Instantaneous Curve Kinematics

The general numerical technique used here to solve the instantaneous, or infinitesimal, inverse kinematics of hyper-redundant manipulators follows from elementary mathematical principles. Furthermore, it is modular enough that it can be implemented in parallel on either digit or analog computers.

Generalizing Equations (1-4), the position and orientation of the distal end of the backbone curve with respect to the base is written in the form:

$$\vec{x}(1,t) = \int_0^1 \vec{v}(\vec{\theta}(s,t))ds \tag{18}$$

and

$$\vec{\Phi}(1,t) = \vec{V}(\vec{\theta}(1,t)), \tag{19}$$

where $\vec{\Phi}(1,t)$ is a representation of spatial rotation. Together, the vector representing end-effector position and orientation is $\vec{x}_{ee}(t) = [\vec{x}^T(1,t), \vec{\Phi}^T(1,t)]^T$. By restricting $\vec{\theta}(s,t)$ to the form in (6), the configuration is a function of $\vec{\mu}$.

The corresponding rate linearized, or differential, kinematics commonly used in robotics can then be written symbolically as:

$$\frac{\partial \vec{x}}{\partial t}(1,t) = \left[\int_0^1 \frac{\partial \vec{v}}{\partial \vec{\theta}} \frac{\partial \dot{\vec{\theta}}(s,\vec{\mu})}{\partial \vec{\mu}} ds\right] \frac{d\vec{\mu}}{dt} \tag{20}$$

and

$$\frac{\partial \vec{\Phi}}{\partial t}(1,t) = \left[\frac{\partial \vec{V}}{\partial \vec{\theta}} \frac{\partial \dot{\vec{\theta}}(1,\vec{\mu})}{\partial \vec{\mu}}\right] \frac{d\vec{\mu}}{dt}. \tag{21}$$

Together these equations can be put in the form of (14), and can be solved for $d\vec{\mu}/dt$ as in (16). The only problem is that if $\dot{\vec{\theta}}(s,\vec{\mu})$ is not an explicit function, indirect methods of computing (15) must be used.

Aside from explicit representation of the vector function $\vec{\theta}(s,\vec{\mu})$, it may be defined by a system of differential equations with initial conditions as in (7) and (8). Without loss of generality, these equations are rewritten in the form:

$$\ddot{\vec{\theta}} + \vec{f}(\dot{\vec{\theta}}, \vec{\theta}, \vec{\mu}, s) = \vec{0} \tag{22}$$

with initial conditions

$$\vec{g}(\dot{\vec{\theta}}(0,t), \vec{\theta}(0,t), \vec{\mu}) = \vec{0} \tag{23}$$

where $\vec{\theta}, \vec{f}(\cdot) \in \mathrm{R}^{\mathrm{M}}$, $\vec{\mu} \in \mathrm{R}^{\mathrm{N}}$, and $\vec{g}(\cdot) \in \mathrm{R}^{2\mathrm{M}}$. M is the minimum number of intrinsic geometric functions needed to fully specify the hyper redundant-manipulator configuration, and N is the number of end-effector coordinates. In the plane $M = 2$ and $N = 3$, while for spatial manipulators $M = 4$ and $N = 6$.

In situations described by (22) and (23) (which is commonly the case when seeking configurations derived from variational problems), a system of *auxiliary* differential equations must be solved in order to generate (20) and (21). These N sets of auxiliary equations are derived by taking the derivatives of (22) and (23) with respect to the N components of $\vec{\mu}$, denoted μ_i. The set of auxiliary equations are written symbolically as the $N \times M$ matrix equation :

$$\frac{d^2}{ds^2}\left(\frac{\partial \vec{\theta}}{\partial \vec{\mu}}\right) + \frac{\partial \vec{f}}{\partial \dot{\vec{\theta}}} \frac{d}{ds}\left(\frac{\partial \vec{\theta}}{\partial \vec{\mu}}\right) + \frac{\partial \vec{f}}{\partial \vec{\theta}} \frac{\partial \vec{\theta}}{\partial \vec{\mu}} = -\frac{\partial \vec{f}}{\partial \vec{\mu}}. \tag{24}$$

Note the linearity of the above equations in the auxiliary variables $\frac{\partial \theta_i}{\partial \mu_j}$ for $(i,j) \in (1,\ldots,M) \times (1,\ldots,N)$. The initial conditions are written symbolically as the $2M \times N$ matrix equation:

$$\left[\frac{\partial \vec{g}}{\partial \dot{\vec{\theta}}} \frac{d}{ds}\left(\frac{\partial \vec{\theta}}{\partial \vec{\mu}}\right) + \frac{\partial \vec{g}}{\partial \vec{\theta}} \frac{\partial \vec{\theta}}{\partial \vec{\mu}} + \frac{\partial \vec{g}}{\partial \vec{\mu}}\right]_{s=0} = 0 \tag{25}$$

which can generally be separated. Note that because derivatives of smooth functions commute,

$$\frac{\partial \dot{\vec{\theta}}}{\partial \mu_i} = \frac{d}{ds}\left(\frac{\partial \vec{\theta}}{\partial \mu_i}\right).$$

Similarly, differentiation and function evaluation commute in the following cases :

$$\frac{\partial \vec{\theta}(0,\mu)}{\partial \mu_i} = \left[\frac{\partial \vec{\theta}(s,\mu)}{\partial \mu_i}\right]_{s=0} \quad \frac{\partial \dot{\vec{\theta}}(0,\mu)}{\partial \mu_i} = \left[\frac{d}{ds}\left(\frac{\partial \vec{\theta}(s,\mu)}{\partial \mu_i}\right)\right]_{s=0}. \tag{26}$$

The simultaneous (possibly parallel) solution of the original system of equations and the auxiliary equations provide the means by which the instantaneous end-effector kinematics of the hyper-redundant manipulator backbone curve is computed at each time step.

While this method may seem very computationally intensive, there are several ways to speed things up. First, if the algorithm is parallelized as in Figure 2, the computation time is no greater than if only the original system of differential equations is integrated forward. Since this must be done for the forward kinematics anyway, there is no loss in time to compute instantaneous inverse kinematics when performed in parallel. Second, if the fastest possible numerical integration techniques are used, or analog implementation of the equations is considered, the solution to initial value problems like (22-23) can be solved approximately in very little time. Third, if computations must be performed at greater speeds than possible using this method, then this method can be used off-line to initiate neural networks or look-up tables which contain the inverse kinematic mapping.

4 Examples

In this section, various examples of the general formulation presented in the previous section are examined. Sections 4.1 - 4.2 consider several examples of nonextensible ($\epsilon = 0$) planar problems. For convenience of notation, all explicit time dependence is suppressed in these problems because time does not enter into the computations in each instantaneous boundary value problem.

4.1 A Linear ODE

Consider the linear ordinary differential equation of second order of the form:

$$\ddot{\kappa} + P(s)\dot{\kappa} + Q(s)\kappa = 0 \tag{27}$$

where $P, Q \in \mathrm{C}^1$, with initial conditions:

$$\kappa(0) = \kappa_0 \quad \dot{\kappa}(0) = \dot{\kappa}_0. \tag{28}$$

It is guaranteed that solutions to this equation are of the 'modal' form :

$$\kappa(s) = a_1\phi_2(s) + a_2\phi_2(s) \tag{29}$$

where ϕ_i for $i = 1,2$ are linearly independent functions such that :

$$\ddot{\phi}_i + P(s)\dot{\phi}_i + Q(s)\phi_i = 0, \tag{30}$$

and a_1, a_2 are constants such that:

$$\begin{pmatrix} \phi_1(0) & \phi_2(0) \\ \dot{\phi}_1(0) & \dot{\phi}_2(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \kappa_0 \\ \dot{\kappa}_0 \end{pmatrix}. \tag{31}$$

Without loss of generality, solutions can be normalize and combined so that the matrix in (31) is the identity, i.e., $a_1 = \kappa_0$ and $a_2 = \dot{\kappa}_0$. Thus, independent of whether or not (27) can be solved in closed form, the solution will be of the modal form in (29) (with $N = 2$ in the case of a second order differential equation, such as (27)). The variables $\{a_i\}$ serve as the free parameters $\{\mu_i\}$ in the case of linear equations. In this special case, $\{a_i\}$ are called the *modal participation factors* because they describe how much each linearly independent solution participates in solving the initial value problem.

In order to generate the partial derivatives :

$$\frac{\partial \theta}{\partial a_i} = \int_0^s \frac{\partial \kappa}{\partial a_i} ds, \tag{32}$$

which are needed to compute the Jacobian matrix (15), the quantities $\frac{\partial \kappa}{\partial a_i}$ for $i = 1, 2$ can be computed by solving the set of auxiliary differential equations:

$$\frac{d^2}{ds^2}\left(\frac{\partial \kappa}{\partial a_i}\right) + P(s)\frac{d}{ds}\left(\frac{\partial \kappa}{\partial a_i}\right) + Q(s)\frac{\partial \kappa}{\partial a_i} = 0 \qquad (33)$$

with initial conditions :

$$\frac{\partial \kappa}{\partial a_1}(0) = 1 \quad \frac{d}{ds}\left(\frac{\partial \kappa}{\partial a_1}\right)(0) = 0 \qquad (34)$$

and

$$\frac{\partial \kappa}{\partial a_2}(0) = 0 \quad \frac{d}{ds}\left(\frac{\partial \kappa}{\partial a_2}\right)(0) = 1. \qquad (35)$$

The notation $F(0)$ is shorthand for the evaluation of any function $F(s,t)$ at $s = 0$.

As a concrete example, consider the most well known of all differential equations :

$$\ddot{\kappa} + \omega^2 \kappa = 0 \qquad (36)$$

with $\omega = 2\pi$, and initial conditions

$$\kappa(0) = a_1 \quad \dot{\kappa}(0) = a_2. \qquad (37)$$

The solution to this equation is of course:

$$\kappa(s) = a_1 \cos 2\pi s + a_2 \frac{\sin 2\pi s}{2\pi}, \qquad (38)$$

which means that

$$\theta(s) = a_1 \sin 2\pi s + a_2 \frac{(1 - \cos 2\pi s)}{2\pi}. \qquad (39)$$

As was shown in [ChB90a,Ch92], the end-effector position (evaluation of (1-2) at $s = 1$) can be written in closed form as :

$$x_{ee} = x_1(0) = \sin(\hat{a}_2) J_0\left[(a_1^2 + \hat{a}_2^2)^{\frac{1}{2}}\right] \qquad (40)$$

$$y_{ee} = x_2(0) = \cos(\hat{a}_2) J_0\left[(a_1^2 + \hat{a}_2^2)^{\frac{1}{2}}\right]$$

where $\hat{a}_2 = \frac{a_2}{2\pi}$.

The "inverse kinematics" (evaluation of modal participation factors) in this case can also be computed in closed form :

$$a_1 = \pm\left(\left[J_0^{-1}\left[(x_{ee}^2 + y_{ee}^2)^{\frac{1}{2}}\right]\right]^2 - [\mathrm{Atan2}(x_{ee}, y_{ee})]^2\right)^{\frac{1}{2}} \qquad (41)$$

$$a_2 = 2\pi \mathrm{Atan2}(x_{ee}, y_{ee}). \qquad (42)$$

J_0^{-1} is the "restricted inverse Bessel function of zero order," and is defined as the inverse of $J_0(x)$ for $0 < x < \mu$ where $\mu \approx 3.832$ is the first local minimum of J_0.

But let us pretend that for some reason we were unable to explicitly represent the mode functions. Then it would not be possible to proceed to find a closed form solution. We would have to use the general techniques of Section 3, or in particular solve the two auxiliary differential equations :

$$\frac{d^2}{ds^2}\left(\frac{\partial \kappa}{\partial a_i}\right) + \omega^2 \frac{\partial \kappa}{\partial a_i} = 0 \qquad (43)$$

for $i = 1, 2$, with initial conditions (34) and (35).

Because these equations are completely independent, they can be solved in parallel using either digital or analog computers. Note that if $\omega \neq n\pi$ for some integral n, closed form forward and inverse kinematic solutions such as (40-42) do not exist, and the problem would have to be solved numerically.

4.2 A Nonlinear ODE

As was mentioned in Subsection 1.2, when seeking backbone curve shapes based on optimization criteria, nonlinear differential equations in the intrinsic functions often arise. See [ChB92a,Ch92] for details, or the appendix for a brief review of how the Euler-Lagrange equations are used. This section shows how these nonlinear differential equations are dealt with using the general formulation in Section 3.

4.2.1 A Straightforward Computation

Suppose that the variables μ_1, μ_2, μ_3 map to the end-effector position and orientation through the differential equation :

$$\ddot{\theta} - \mu_1 \cos\theta + \mu_2 \sin\theta = 0 \qquad (44)$$

with initial conditions :

$$\theta(0) = 0 \quad \dot{\theta}(0) = \mu_3. \qquad (45)$$

Auxiliary equations are generated by simply taking derivatives of the differential equation and initial conditions with respect to the variables μ_1, μ_2, μ_3.

This results in the differential equations :

$$\frac{d^2}{ds^2}\left(\frac{\partial \theta}{\partial \mu_1}\right) + \mu_1 \frac{\partial \theta}{\partial \mu_1}\sin\theta + \mu_2 \frac{\partial \theta}{\partial \mu_1}\cos\theta = \cos\theta \qquad (46)$$

with

$$\frac{\partial \theta}{\partial \mu_1}(0) = \frac{d}{ds}\left(\frac{\partial \theta}{\partial \mu_1}\right)(0) = 0, \qquad (47)$$

$$\frac{d^2}{ds^2}\left(\frac{\partial \theta}{\partial \mu_2}\right) + \mu_1 \frac{\partial \theta}{\partial \mu_2}\sin\theta + \mu_2 \frac{\partial \theta}{\partial \mu_2}\cos\theta = -\sin\theta \qquad (48)$$

with

$$\frac{\partial \theta}{\partial \mu_2}(0) = \frac{d}{ds}\left(\frac{\partial \theta}{\partial \mu_2}\right)(0) = 0, \qquad (49)$$

and

$$\frac{d^2}{ds^2}\left(\frac{\partial \theta}{\partial \mu_3}\right) + \mu_1 \frac{\partial \theta}{\partial \mu_3}\sin\theta + \mu_2 \frac{\partial \theta}{\partial \mu_3}\cos\theta = 0 \qquad (50)$$

with

$$\frac{\partial \theta}{\partial \mu_3}(0) = 0 \quad \frac{d}{ds}\left(\frac{\partial \theta}{\partial \mu_3}\right)(0) = 1. \qquad (51)$$

Each of these initial value problems can be solved separately by integrating forward simultaneously with (44-45). Note that these differential equations are linear in the variables $\frac{\partial \theta}{\partial \mu_i}$, with nonconstant coefficients and forcing terms which are dependent on θ.

Figure 3 shows configurations defined by the Euler-Lagrange equations which have the end-effector follow the trajectory $(x_{ee}, y_{ee}) = (t + \frac{1}{4}, \frac{1}{2})$ for $t \in [0, \frac{1}{2}]$. In this example, since end-effector position is the only quantity of interest, the constraint $\dot{\theta}(0) = 0.2 = \mu_3$ is imposed arbitrarily. In this way, the manipulator effectively acts as if it only has two degrees of freedom. Thus, the Jacobian for this case is a 2×2 matrix.

Figure 4 shows the workspace corresponding to the configurations generated by (44-45) for $(\mu_1, \mu_2) \in [-18, 18] \times [-15, 55]$. Figures 5(a)-(b) respectively show where configurations are singular in the μ_1-μ_2 space and the workspace. Values of μ_1 and μ_2 which cause the Jacobian to have determinants with absolute value less than 0.05 are considered singular. In Figure 5(a), the light region indicates where the Jacobian is nonsingular, whereas in Figure 5(b) the dark region indicates where the Jacobian is nonsingular.

4.2.2 Transforming Equations into Simpler Forms

Equations (44-45) can easily be transformed into a simpler form. Taking the derivative of (44) with respect to s yields:

$$\frac{d^3\theta}{ds^3} + (\mu_1 \sin\theta + \mu_2 \cos\theta)\frac{d\theta}{ds} = 0 \tag{52}$$

with resulting initial conditions :

$$\theta(0) = 0 \quad \dot{\theta}(0) = \mu_3 \quad \ddot{\theta}(0) = \mu_1. \tag{53}$$

Multiplying (52) by $\dot{\theta}$, integrating, and evaluating with initial conditions (53), yields

$$\frac{1}{2}\dot{\theta}^2 - (\mu_1 \sin\theta + \mu_2 \cos\theta) = \frac{1}{2}\mu_3^2 - \mu_2. \tag{54}$$

Combining (52) and (54) so as to eliminate the trigonometric terms:

$$\ddot{\kappa} + \kappa\left(\frac{1}{2}\kappa^2 + \mu_2 - \frac{1}{2}\mu_3^2\right) = 0 \tag{55}$$

with initial conditions :

$$\kappa(0) = \mu_3 \quad \dot{\kappa}(0) = \mu_1. \tag{56}$$

θ can then be computed from :

$$\theta(s,t) = \int_0^s \kappa(\sigma,t)d\sigma. \tag{57}$$

A change of coordinates for the free parameters can be defined as $\mu_2 - \frac{1}{2}\mu_3^2 = q_3$, $\mu_1 = q_2$, and $\mu_3 = q_1$, to simplify the equations. The decision as to what form the equations and free parameters should be manipulated into depends on how the solution will be implemented. In other words, some forms are more amenable to numerical solution because a reduced number of arithmetic operations and function calls need to be performed, while other forms may be more convenient for analog computer implementation because of particular hardware features.

Thus, we can rewrite (55-56) as:

$$\ddot{\kappa} + \kappa(\frac{1}{2}\kappa^2 - q_3) = 0 \tag{58}$$

with initial conditions

$$\kappa(0) = q_1 \quad \dot{\kappa}(0) = q_2. \tag{59}$$

The corresponding auxiliary equations are:

$$\frac{d}{ds}\left(\frac{\partial\kappa}{\partial q_i}\right) + \left(\frac{3}{2}\kappa^2 - q_3\right)\frac{\partial\kappa}{\partial q_i} = 0 \tag{60}$$

for $i = 1, 2$ with boundary conditions:

$$\frac{\partial\kappa}{\partial q_1}(0) = \frac{d}{ds}\left(\frac{\partial\kappa}{\partial q_2}\right)(0) = 1 \tag{61}$$

and

$$\frac{\partial\kappa}{\partial q_2}(0) = \frac{d}{ds}\left(\frac{\partial\kappa}{\partial q_1}\right)(0) = 0. \tag{62}$$

The third auxiliary equation is :

$$\frac{d}{ds}\left(\frac{\partial\kappa}{\partial q_3}\right) + \left(\frac{3}{2}\kappa^2 - q_3\right)\frac{\partial\kappa}{\partial q_3} = \kappa \tag{63}$$

with initial conditions

$$\frac{\partial\kappa}{\partial q_3}(0) = \frac{d}{ds}\left(\frac{\partial\kappa}{\partial q_3}\right)(0) = 0 \tag{64}$$

From the solution of these equations, we can then compute :

$$\frac{\partial\theta}{\partial q_i} = \int_0^s \frac{\partial\kappa}{\partial q_i}ds. \tag{65}$$

5 Alternate Formulations

Alternatives to the intrinsic formulation and associated auxiliary equations are examined in this section. In Subsection 5.1 an *extrinsic* formulation of the same problem examined in Section 4.2 is introduced. In Subsection 5.2 it is shown how the shooting technique, which is applicable to both intrinsic and extrinsic formulations, is achieved without auxiliary equations.

5.1 An Extrinsic Formulation

This section illustrates how *extrinsic* parameterizations can be used to reformulate the problem considered in the previous section. An extrinsic formulation uses Cartesian coordinates to define a curve and imposes appropriate constraints. This contrasts to the intrinsic approach used previously in this paper, in which a minimal number of geometric functions where used to define the backbone curve shape (and thus the manipulator configuration). The benefits of the extrinsic formulation is that for some problems, intrinsic formulations lead to intractable mathematical problems.

As an example of an extrinsically formulated problem, the same extremal bending criterion defined for the previous example is used. However, the algebraic description of the Euler-Lagrange equations and the related numerical solutions are significantly different.

In the extrinsic formulation, the measure of total curve bending is defined as

$$I = \int_0^1 \ddot{\vec{x}} \cdot \ddot{\vec{x}}\, ds. \tag{66}$$

This is equivalent to (10) according to the definition of curvature when $\epsilon(s,t) = 0$. This can be verified by substituting (1-3) into (66). A benefit of the extrinsic formulation is that there is no need for the isoperimetric constraints like (11). The drawback however, is that algebraic constraints of the form :

$$\dot{\vec{x}} \cdot \dot{\vec{x}} = 1 \tag{67}$$

must be imposed to ensure that the curve parameter is arc length in the nonextensible case. Otherwise, the quantity in (66) is not a measure of pure bending, because the physical meaning of the curve parameter varies with configuration.

Therefore, defining

$$\mathcal{L}(s) = \frac{1}{2}(\ddot{\vec{x}} \cdot \ddot{\vec{x}}) + \mu_v(s)(\dot{\vec{x}} \cdot \dot{\vec{x}} - 1) \tag{68}$$

and using the Euler-Lagrange equation as explained in the appendix, a system of differential equations result which are written in state-space form as:

$$\mathbf{H}(\vec{X})\dot{\vec{X}} = \vec{f}(\vec{X}), \tag{69}$$

where $\vec{X} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9]^T$, $x_{i+2} = \frac{dx_i}{ds}$, $x_{i+4} = \frac{d^2x_i}{ds^2}$, and $x_{i+6} = \frac{d^3x_i}{ds^3}$ for $i \in \{1,2\}$, and $x_9 = \mu_v$. Also:

$$\mathbf{H}(\vec{X}) = \begin{pmatrix} \mathbf{I}_6 & 0 \\ 0 & \mathbf{A} \end{pmatrix} \tag{70}$$

where $\mathbf{I}_6$ is the 6×6 identity matrix,

$$\mathbf{A}(\vec{X}) = \begin{pmatrix} 1 & 0 & -x_3 \\ 0 & 1 & -x_4 \\ x_3 & x_4 & 0 \end{pmatrix}, \tag{71}$$

and

$$\vec{f}(\vec{X}) = \begin{pmatrix} \vec{g}(\vec{X}) \\ \vec{h}(\vec{X}) \end{pmatrix}, \tag{72}$$

where the components of $\vec{g}(\vec{X})$ are

$$g_i(\vec{X}) = x_{i+2} \tag{73}$$

for $i \in [1, 6]$ and

$$\vec{h}(\vec{X}) = \begin{pmatrix} x_9 x_5 \\ x_9 x_6 \\ -3(x_5 x_7 + x_6 x_8) \end{pmatrix} \tag{74}$$

The matrix $\mathrm{H}(\vec{X})$ is inverted to yield:

$$\dot{\vec{X}} = \mathrm{H}^{-1}(\vec{X})\vec{f}(\vec{X}) \tag{75}$$

where

$$\mathrm{H}^{-1}(\vec{X}) = \begin{pmatrix} \mathrm{I}_6 & 0 \\ 0 & \mathrm{A}^{-1} \end{pmatrix}, \tag{76}$$

and

$$\mathrm{A}^{-1}(\vec{X}) = \begin{pmatrix} x_4^2 & -x_3 x_4 & x_3 \\ -x_3 x_4 & x_3^2 & x_4 \\ -x_3 & -x_4 & 1 \end{pmatrix}. \tag{77}$$

Equation (75) has initial conditions: $x_1(0) = x_2(0) = x_3(0) = x_6(0) = 0$; $x_4(0) = 1$; $x_5(0) = \mu_1$; $x_7(0) = \mu_2$; $x_8(0) = -\mu_1^2$; and $x_9(0) = \mu_3$. $\{\mu_1, \mu_2, \mu_3\}$ are undetermined constants which are consistent with the constraints at $s = 0$ (such as $\dot{\vec{x}}(0) \cdot \dot{\vec{x}}(0) = 1$), and the derivatives of the constraint equation, e.g., $\dot{\vec{x}} \cdot \dot{\vec{x}} = 1$ implies $\dot{\vec{x}} \cdot \ddot{\vec{x}} = 0$ for all s including $s = 0$. Because $\vec{x}$ is a function of $\{\mu_i\}$, we write

$$\vec{x}(s, t) = \hat{\vec{x}}(s, \vec{\mu}(t)). \tag{78}$$

The values of $\{\mu_i\}$ must be chosen to satisfy end-effector position constraints. (75) is solved numerically using the same kind of shooting method developed for the intrinsically formulated problem. That is, the Jacobian matrix is written symbolically as

$$\mathrm{J}(\vec{\mu}) = \frac{\partial \hat{\vec{x}}}{\partial \vec{\mu}} \tag{79}$$

where the collums of the Jacobian are generated directly by integrating:

$$\frac{d}{ds}\left(\frac{\partial \vec{X}}{\partial \vec{\mu}}\right) = \frac{\partial}{\partial \vec{\mu}}\left[H^{-1}(\vec{X})\vec{f}(\vec{X})\right] \tag{80}$$

with respect to s.

Both the intrinsic and extrinsic formulations have advantages and drawbacks. For instance, an advantage of the extrinsic approach is that the auxiliary equations are all of the same form, with differences occurring only in the initial conditions. This is a benefit for modular implementation and ease in programming because the same function can be called with different boundary conditions. Another benefit of the extrinsic formulation is the relative ease with which complicated variational problems can be modeled. On the other hand, the intrinsic formulation has the benefit of a reduced number of equations which must be computed, and provides a more easily visualized description of the geometry of the backbone curve.

5.2 Approximate Shooting without Auxiliary Equations

In either the intrinsic or extrinsic formulations, explicit computation of auxiliary equations can be avoided by simply perturbing the reduced set of free variables and using the definition of the partial derivative. That is,

$$\frac{\partial \vec{x}(s, \vec{\mu})}{\partial \mu_i} \approx \frac{1}{\nu}\left[\vec{x}(s, \vec{\mu} + \nu \vec{e}_i) - \vec{x}(s, \vec{\mu})\right] \tag{81}$$

for $0 < \nu << 1$. $\{\vec{e}_i\}$ are the natural basis vectors for R^{N}. In this way no auxiliary equations need to be computed. However, the potential for numerical ill-conditioning increases.

6 Conclusions

Methods for generating the inverse kinematics of hyper-redundant manipulator configurations without explicitly defined intrinsic shape functions have been presented. Examples illustrated the technique and showed how parallel numerical algorithms can be implemented. In addition, algorithmic singularities associated with a special class of constraints on hyper-redundant manipulator configurations were examined.

7 Appendix : Review of the Calculus of Variations

The essential results from the calculus of variations which are used in [ChB92a,Ch92] to derive some of the differential equations used as examples in Sections 4 and 5 are summarized here. Equations (10-11) and (66-67) were used to generate manipulator configurations. These equations are of the general form:

$$I = \int_0^1 f(s, \vec{\theta}(s), \vec{\theta}^1(s), \cdots, \vec{\theta}^n(s))ds. \tag{82}$$

$\vec{\theta}(s) \in \mathrm{R}^{\mathrm{N}}$ is a set of functions such as the backbone curve intrinsic functions or the extrinsic coordinates of points on the curve. $f(\cdot)$ is a physically motivated function, e.g., curvature squared. $\vec{\theta}^i$ is shorthand for $\vec{\theta}^i = \frac{d^i\vec{\theta}}{ds^i}$. [ChB92a] dealt with extremizing (82) subject to *integral* constraints (which arise from end-effector position constraints) of the form:

$$\int_0^1 \vec{g}(s, \vec{\theta}(s), \vec{\theta}^1(s), \cdots, \vec{\theta}^n(s))ds = \vec{x}_D. \tag{83}$$

(82) may also be subject to *finite* constraints of the form:

$$\vec{h}(s, \vec{\theta}(s), \vec{\theta}^1(s), \ldots, \vec{\theta}^n(s)) = \vec{0}, \tag{84}$$

as are encountered in the extrinsic formulation. The calculus of variations [Ew69] provides a means for finding a $\vec{\theta}(s)$ which yields extremal values of Equation (82) with constraints (83) and/or (84). To solve such problems, define a function (which we will call the Lagrangian):

$$\mathcal{L} = f + \vec{\mu}_c \cdot \vec{g} + \vec{\mu}_v \cdot \vec{h},$$

where $\vec{\mu}_c$ and $\vec{\mu}_v(s)$ are respectively constant and variable Lagrange multipliers associated with the isoperimetric and finite constraints. The $\vec{\theta}(s)$ which are extremals of (82) subject to constraints (83) or (84) is a solution to the Euler-Lagrange equations:

$$\sum_{i=0}^{n}(-1)^i \frac{d^i}{ds^i}\left(\frac{\partial \mathcal{L}}{\partial \theta_j^i}\right) = 0 \quad j = 1, \cdots, N. \tag{85}$$

With constraints (83) or (84) and boundary conditions $\vec{\theta}^i(0) = \vec{\theta}_0^i$ and $\vec{\theta}^i(1) = \vec{\theta}_1^i$ for $i \in [0, 1, \ldots, n]$, (85) can be solved to find the extremals, $\vec{\theta}$, and Lagrange multipliers $\vec{\mu}_c$ and $\vec{\mu}_v(s)$. Necessary conditions for solutions to (85) is discussed in [Ew69], while sufficiency conditions for minimal solutions can be found in [Bre91].

8 References

[Bre91] Brechtken-Manderscheid, U., *Introduction to the Calculus of Variations*, Chapman and Hall, New York, 1991.

[Ch92] Chirikjian, G.S., "Theory and Applications of Hyper-Redundant Robotic Manipulators," Ph.D thesis, Dept. of Applied Mechanics, California Institute of Technology, June, 1992.

[ChB90a] Chirikjian, G.S., Burdick, J.W., "An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators," *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinatti, OH, May 13-18, 1990

[ChB90b] 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.

[ChB91a] 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.

[ChB91b] Chirikjian, G.S., Burdick, J.W., "Kinematics of Hyper-Redundant Locomotion with Applications to Grasping," *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, April, 1991.

[ChB91c] Chirikjian, G.S., Burdick, J.W., "Theory and Applications of Hyper-Redundant Robotic Mechanisms," 8^{th} *World Congress on the Theory of Machines and Mechanisms*, Prague, Czechoslovakia, August 1991.

[ChB92a] Chirikjian, G.S. and Burdick, J.W., " Kinematically Optimal Hyper-Redundant Manipulator Configurations," *IEEE Conference on Robotics and Automation*, Nice, France, May, 1992.

[Ew69] Ewing, G.M., *Calculus of Variations With Applications*, W.W. Norton and Co. Inc., New York, 1969.

[Ke68] Keller, H.B., *Numerical Methods for Two-Point Boundary-Value Problems*, Blaisdell Publishing Co., Waltham Mass., 1968.

[KuH83] Kubicek, M., Hlavacek, V., *Numerical Solution of Nonlinear Boundary Value Problems with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983

[Me73] Meyer, G.H., *Initial Value Methods for Boundary Value Problems*, Academic Press, New York, 1973.

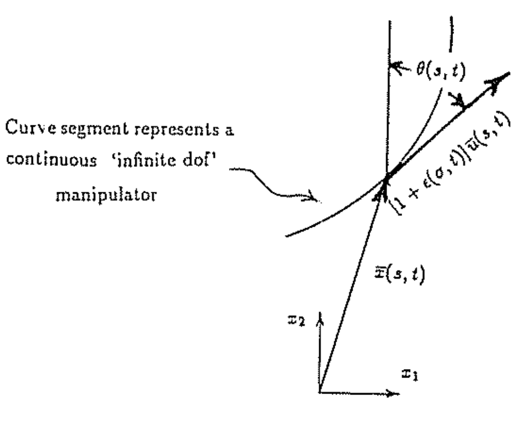

Figure 1: A Continuous 'Infinitely Redundant' Manipulator

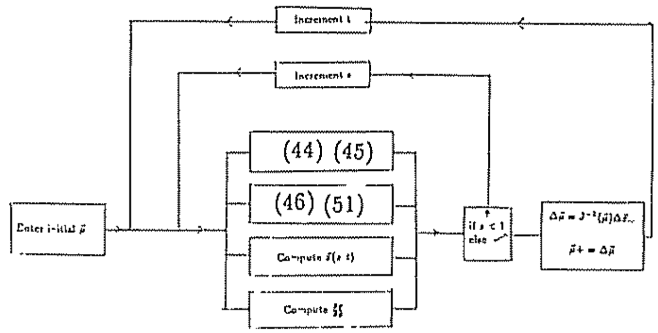

Figure 2: Schematic of a Parallel Algorithm

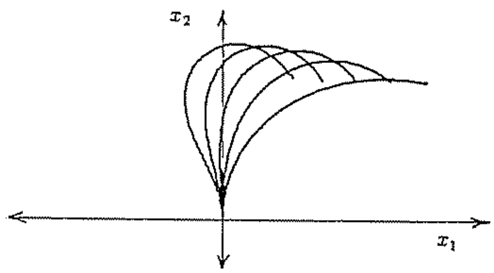

Figure 3: Configurations Associated with Equations (44-45)

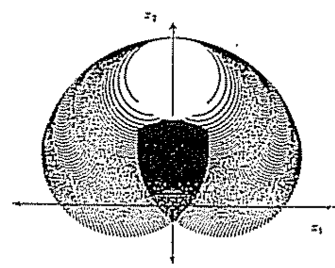

Figure 4: Workspace Associated with Equations (44-45)

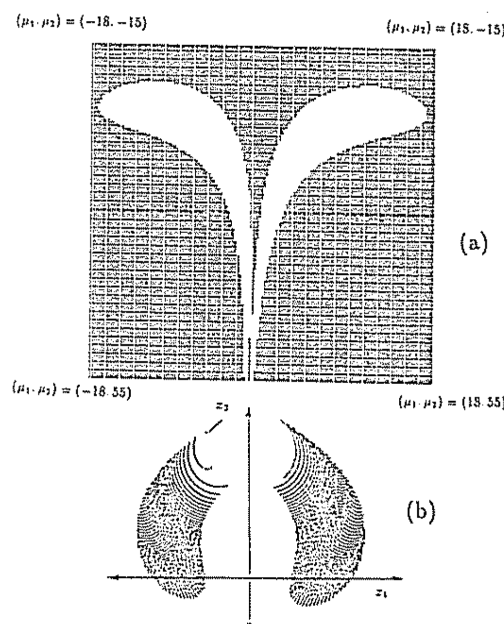

Figure 5: Singularities Associated with Equations (44-45)