

A General Numerical Method for Hyper-Redundant Manipulator Inverse Kinematics

Gregory S. Chirikjian

Dept. of Mechanical Engineering
Johns Hopkins University
Baltimore, MD 21218

Abstract

"Hyper-redundant" robots have a very large or infinite degree of kinematic redundancy. This paper introduces a new generalized resolved rate technique for solving hyper-redundant manipulator inverse kinematics using a 'backbone curve.' This method is 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 which were previously intractable can now be handled easily with this technique. Examples include configurations generated using the calculus of variations. Also, the method is naturally parallelizable for fast digital and/or analog computation.

1 Introduction

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.

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 technique for hyper-redundant manipulator inverse kinematics.

1.1 Review of Hyper-Redundant Manipulator 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

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

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

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

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

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

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

$K(s,t)$ 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:

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

In addition to this idealized infinite degree of freedom case, (1)-(5) are used to define continuous ‘backbone curves’ [ChB90b] for discrete hyper-redundant manipulator morphologies 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 :

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

for $\vec{\mu} \in \mathbf{R}^N$. 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 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 $\vec{\theta}(s, \vec{\mu}(t))$ is not available. In some situations only systems of differential equations of the form:

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

with initial conditions

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

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

1.2 Some Motivational Examples

In previous papers, techniques for resolving hyper-redundancy were presented [ChB90b, ChB92a]. 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 *inextensible* manipulators, i.e., manipulators with $\epsilon(s, t) = 0$, is

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

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

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

subject to the end-effector constraints:

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

Differential equations such as:

$$\ddot{\theta} - \mu_1 \cos \theta + \mu_2 \sin \theta = 0 \quad (12)$$

together with integral constraints (11) and initial conditions

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

result when using the calculus of variations (see [ChB92a] for a derivation). Here, and throughout this paper, a $\dot{\cdot}$ 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 :

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

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

$$\begin{pmatrix} \int_0^1 \frac{\partial \theta}{\partial \mu_1} \cos \theta ds & \int_0^1 \frac{\partial \theta}{\partial \mu_2} \cos \theta ds & \int_0^1 \frac{\partial \theta}{\partial \mu_3} \cos \theta ds \\ -\int_0^1 \frac{\partial \theta}{\partial \mu_1} \sin \theta ds & -\int_0^1 \frac{\partial \theta}{\partial \mu_2} \sin \theta ds & -\int_0^1 \frac{\partial \theta}{\partial \mu_3} \sin \theta ds \\ \frac{\partial \theta}{\partial \mu_1} & \frac{\partial \theta}{\partial \mu_2} & \frac{\partial \theta}{\partial \mu_3} \end{pmatrix}. \quad (15)$$

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

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

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

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

(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).

2 Standard Techniques for Numerical Solution of 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 Rate-Linearized Inverse Kinematics of Backbone Curves

The general numerical technique used here to solve the differential inverse kinematics of hyper-redundant manipulators follows from elementary mathematical principles.

From Equations (1)-(4), the position and orientation of the distal end of the backbone curve with respect to the base will be of the form:

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

and

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

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:

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

and

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

Strictly speaking, differentiation with respect to a vector is not mathematically correct notation. However,

such notation is understood to mean differentiation with respect to each component of the vector.

Together (20-21) 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 $\vec{\theta}(s, \vec{\mu})$ is not an explicit function, indirect methods of computing (20) and (21) 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 of the form :

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

with initial conditions

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

where $\vec{\theta}, \vec{f}(\cdot) \in \mathbf{R}^M$, $\vec{\mu} \in \mathbf{R}^N$, and $\vec{g}(\cdot) \in \mathbf{R}^{2M}$. M is the minimum number of intrinsic geometric functions needed to fully specify the hyper-redundant manipulator's backbone curve 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. 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 :

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

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

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

which can generally be separated. Note that because derivatives of smooth functions with respect to independent variables commute,

$$\frac{\partial \vec{\theta}}{\partial \mu_i} = \frac{d}{ds} \left(\frac{\partial \vec{\theta}}{\partial \mu_i} \right). \quad (26)$$

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

$$\frac{\partial \vec{\theta}(0, \mu)}{\partial \mu_i} = \left[\frac{\partial \vec{\theta}(s, \mu)}{\partial \mu_i} \right]_{s=0} \quad (27)$$

$$\frac{\partial \ddot{\vec{\theta}}(0, \mu)}{\partial \mu_i} = \left[\frac{d}{ds} \left(\frac{\partial \vec{\theta}(s, \mu)}{\partial \mu_i} \right) \right]_{s=0}. \quad (28)$$

The simultaneous (possibly parallel) solution of the original system of equations and the auxiliary equations provides 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)-(25) 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.

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. An example now illustrates how these nonlinear differential equations are dealt with using the general formulation presented earlier in this section.

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

$$\ddot{\theta} - \mu_1 \cos \theta + \mu_2 \sin \theta = 0 \quad (29)$$

with initial conditions :

$$\theta(0) = 0 \quad \dot{\theta}(0) = \mu_3. \quad (30)$$

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 :

$$\frac{d^2}{ds^2} \left(\frac{\partial \theta}{\partial \mu_1} \right) + \mu_1 \frac{\partial \theta}{\partial \mu_1} \sin \theta + \mu_2 \frac{\partial \theta}{\partial \mu_1} \cos \theta = \cos \theta \quad (31)$$

with

$$\frac{\partial \theta}{\partial \mu_1}(0) = \frac{d}{ds} \left(\frac{\partial \theta}{\partial \mu_1} \right) (0) = 0, \quad (32)$$

$$\frac{d^2}{ds^2} \left(\frac{\partial \theta}{\partial \mu_2} \right) + \mu_1 \frac{\partial \theta}{\partial \mu_2} \sin \theta + \mu_2 \frac{\partial \theta}{\partial \mu_2} \cos \theta = -\sin \theta \quad (33)$$

with

$$\frac{\partial \theta}{\partial \mu_2}(0) = \frac{d}{ds} \left(\frac{\partial \theta}{\partial \mu_2} \right) (0) = 0, \quad (34)$$

and

$$\frac{d^2}{ds^2} \left(\frac{\partial \theta}{\partial \mu_3} \right) + \mu_1 \frac{\partial \theta}{\partial \mu_3} \sin \theta + \mu_2 \frac{\partial \theta}{\partial \mu_3} \cos \theta = 0 \quad (35)$$

with

$$\frac{\partial \theta}{\partial \mu_3}(0) = 0 \quad \frac{d}{ds} \left(\frac{\partial \theta}{\partial \mu_3} \right) (0) = 1. \quad (36)$$

Each of these initial value problems can be solved separately by integrating forward simultaneously with (29) and (30). 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, and a 2×2 Jacobian is used.

4 Singularities Associated with Complexity Reduction

While constraining intrinsic geometric functions such as $\theta(s, t)$ and $\epsilon(s, t)$ to the form (6) reduces the complexity of end-effector positioning and trajectory tracking, it introduces new algorithmic singularities. Recall that singularities of standard kinematically sufficient manipulators can be detected when the manipulator Jacobian loses rank. Similarly, algorithmic singularities associated with hyper-redundant manipulators with intrinsic functions constrained to the form (6) will occur when

$$\det(\mathbf{J}) = 0. \quad (37)$$

Given explicit functions $\tilde{\theta}(s, \vec{\mu})$, inverse kinematics can sometimes be solved in closed form, as was done in [ChB90b].

Using the methods presented earlier in this paper, (16) can be solved numerically without explicitly being given the functions $\tilde{\theta}(s, \vec{\mu})$.

The easiest method for solving (16) numerically is simply to test a discretized set of all acceptable values of the free parameters. Then check which ones yield singularities by computing $\mathbf{J}(\vec{\mu})$ as described earlier, and take the determinant. Space limitations prevent additional discussion of this issue. See [Ch93] for more information.

5 Conclusions

A new method for generating the inverse kinematics of hyper-redundant manipulator configurations without explicitly defined intrinsic shape functions has been presented. Examples illustrated the technique and showed how parallel numerical algorithms can be implemented.

6 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*, Cincinnati, 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 Mech-

anisms," 8th 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.

[ChB93] Chirikjian, G.S., Burdick, J.W., "Design and Experiments with a 30 DOF Robot," *IEEE Conference on Robotics and Automation*, Atlanta, 1993.

[Ch93] Chirikjian, G.S., "A General Numerical Method for Computing Hyper-Redundant Manipulator Inverse Kinematics," *Technical Report JHU-REMS-93-1*.

[Di94] Dixon, A.C., *The Elementary Properties of the Elliptic Functions*, MacMillan and Co., New York, 1894.

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

[Gru80] Gruver, W.A., Sachs, E., *Algorithmic Methods in Optimal Control*, Pitman Publishing, Ltd., Boston, 1980.

[Kc68] Keller, H.B., *Numerical Methods for Two-Point Boundary-Value Problems*, Blaisdell Publishing Co., Waltham Mass., 1968.

[Ku183] Kubicek, M., Hlavacek, V., *Numerical Solution of Nonlinear Boundary Value Problems with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983

[La89] Lawden, D.F., *Elliptic Functions and Applications*, Springer-Verlag, New York, 1989.

[Mc73] Meyer, G.H., *Initial Value Methods for Boundary Value Problems*, Academic Press, New York, 1973.

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

[TaD90] Tavakkoli, S., Dhande, S.G., "Shape Synthesis and Optimization Using Intrinsic Geometry" *Proc. ASME Design Conference*, Chicago, Il, Sept. 16-19, 1990.

[WiM89] Wilson, J.F., Mahajan, U., "The Mechanics and Positioning of Highly Flexible Manipulator Limbs," *Journal of Mechanisms, Transmissions, and Automation in Design*. Vol, 111, June 1989.

[WiS88] Wilson, J.F., Snyder, J.M., "The Elastica with End-Load Flip-Over," *Journal of Applied Mechanics*. Vol. 55, December 1988.

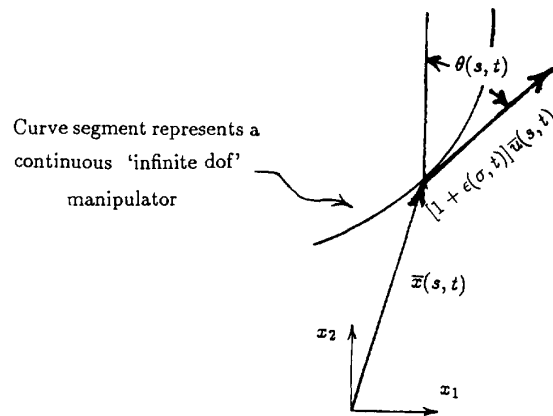


Figure 1: A Continuous 'Infinitely Redundant' Manipulator

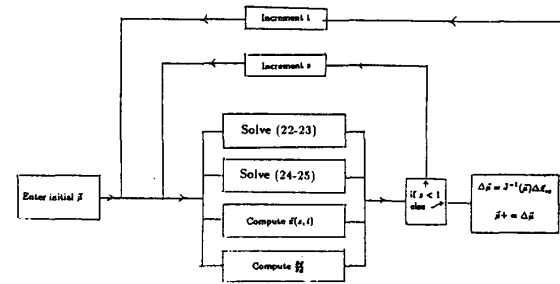


Figure 2: Schematic of a Parallel Algorithm

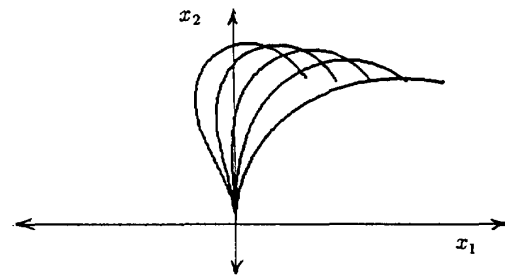


Figure 3:

Configurations Associated with Equations (29-30)