An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators

Gregory S. Chirikjian, Joel W. Burdick

School of Engineering and Applied Science
California Institute of Technology
Pasadena, CA   91125

Abstract

Many tasks which manipulators must perform occur in the presence of obstacles. While a variety of algorithms for non-redundant and mildly redundant manipulators exist, little analysis has been performed for a class of robots with the most to offer in the area of obstacle avoidance: hyper-redundant manipulators. The term "hyper-redundant" refers to redundant manipulators with a very large, possibly infinite, number of degrees of freedom. These manipulators are analogous in morphology and operation to "snakes," "elephant trunks," and "tentacles." This paper presents novel kinematic algorithms for implementing planar hyper-redundant manipulator obstacle avoidance. Unlike artificial potential field methods, the method outlined in this paper is strictly geometric. 'Tunnels' are defined in a workspace in which obstacles are present. Methods of differential geometry are then used to formulate equations which guarantee that sections of the manipulator are confined to the tunnels, and therefore avoid the obstacles. A general formulation is given with examples to illustrate this approach.

1. Introduction

Several methods for dealing with the problem of robot obstacle avoidance in a time-independent workspace environment have been developed by other investigators [1,4,12, and references therein]. One popular method is the artificial potential field [12]. In this method, an artificial repulsive potential field is assumed between the manipulator and obstacles in the workspace. Similarly, in other methods a measure of distance from obstacles to the manipulator can be defined, and optimized to yield configurations in which the manipulator does not touch obstacles [1]. These methods have potential drawbacks. In the artificial potential field approach, the robot can 'get stuck' in a potential well, although some methods to circumvent this problem have been investigated [13]. The drawback of many optimization methods is computational complexity, since many of these methods are based on the computation of a pseudo-inverse Jacobian matrix, which is expensive when the number of degrees of freedom becomes large.

This paper presents a novel approach, termed "tunneling," to the obstacle avoidance problem which is applied to planar hyper-redundant manipulators. A hyper-redundant manipulator is a redundant manipulator in which the number of redundant degrees of freedom is large or infinite. These manipulators are analogous in morphology to snakes or tentacles. A number of recent works have been devoted to the design and systems aspects of snake-like manipulators [9,10,11,16,17], although less work has been done in the kinematic analysis of these manipulators. In [5] general methods for analyzing the kinematics of planar and spatial hyper-redundant manipulators are presented, and relevant details are summarized below.

1.1. Kinematics of Planar Hyper-redundant Manipulators

Hyper-redundant manipulators with constant base to end effector length are referred to throughout this paper as *tensor* manipulators. A planar hyper-redundant tensor manipulator may be comprised of many rigid links, as in Figure 1(a), or the physical construction of the device may be truly flexible—such as a pneumatic or tendon based structure, as in Figure 1(b).
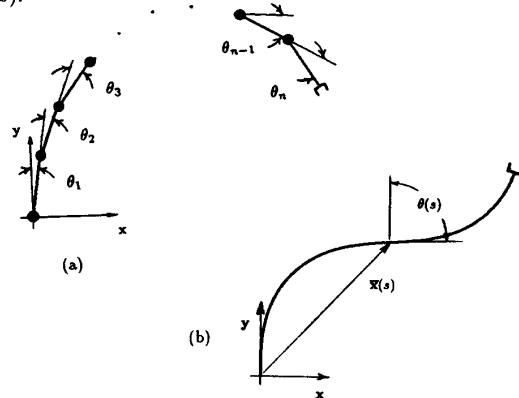

Figure 1: Discrete and Continuous Hyper-Redundant Manipulators

In the case of a truly flexible structure, a "backbone curve" of constant length can be defined which exactly captures the manipulator shape. In the case of many discrete links, the essential macroscopic features of the manipulator can be captured by a backbone curve of the same length as the sum of the link lengths. For analysis purposes, the discrete manipulator is modeled as a continuous curve, and kinematic and obstacle avoidance computations are performed using

the continuous backbone curve. The actual physical manipulator is then fitted to the idealized backbone curve solutions to complete analysis of the kinematics. Consequently, the complete analysis of planar hyper-redundant tensor manipulators can be reduced to the study of the differential geometry of planar curves, with an additional "fitting" process required for manipulators consisting of a finite number of discrete links. We assume that in the discrete structure case the number of links is large so that the manipulator profile does not substantially deviate from a backbone curve with bounded curvature.

1.2 Continuous Hyper-redundant Manipulator Kinematics

Figure (1a) shows one possible discrete hyper-redundant manipulator. Figure (1b) is the 'backbone' curve of a tensor manipulator, which is the locus of all points in the x_1-x_2 plane which have position defined by $\overline{x}(s)$, where s is the arc length measured from the origin. It is assumed that s is in the range $0 \le s \le 1$, meaning all lengths in the plane are scaled to units of the manipulator length. $\theta(s_0)$ is the angle which the tangent to the curve at the point $s = s_0$ makes with the tangent at the point $s = 0$, i.e., the angle it makes with the x_2 axis measured in a clockwise sense.

A curve with arbitrary position and orientation in the plane can be described by the equation:

$$\overline{r}(s) = \mathbf{Q}\overline{x}(s) + \overline{c} \qquad (1)$$

where $\overline{c}$ is a 2×1 vector, and $\mathbf{Q}$ is a 2×2 rotation matrix. $\mathbf{Q}$ and $\overline{c}$ do not define the shape of the curve, only its location in the plane. $\overline{x}(s)$ is a vector of the form $\overline{x}(s) = [x_1(s) \ x_2(s)]^T$. Each component of $\overline{x}(s)$ must satisfy the following equation:

$$\left(\frac{d^2 x_i}{ds^2}\right)^2 = \kappa^2(s)\left[1 - \left(\frac{dx_i}{ds}\right)^2\right] \qquad (2)$$

where $\kappa(s)$ is the curvature function, which is the magnitude of the rate of change of the unit tangent vector at a point s along the curve: $\kappa(s) = |d^2\overline{x}/ds^2|$. Solutions to (2) are of the form:

$$\begin{aligned} x_1(s) &= \int_0^s \sin(\theta(\mu))d\mu \\ x_2(s) &= \int_0^s \cos(\theta(\mu))d\mu \end{aligned} \qquad (3)$$

where:

$$\theta(s) = \int_0^s \kappa(\sigma)d\sigma. \qquad (4)$$

These solutions assume that the base frame of the curve is defined by the unit vectors $\overline{e}_i$ such that $\overline{x}(0) = 0$ and that the tangent to the curve at its base point is $\overline{x}'(0) = \overline{e}_2$, where a ' represents differentiation with respect to s. The rotation matrix, $\mathbf{Q}$, and translation vector, $\overline{c}$, follow from whatever initial conditions are imposed on the governing differential equation. Consequently, for given boundary conditions, the curvature function uniquely specifies the entire shape of the planar curve. $\mathbf{Q}$ and $\overline{c}$ can be functions of time if the base is rotating and translating respectively. This might occur if the

manipulator is mounted on a moving base or if the hyper-redundant mechanism is used for snake-like locomotion [8].

Similarly, the curvature function need not be a function of arc length alone, and could depend upon time as well. The forward kinematics of planar tensor mechanisms is given by (3) and (4), and the inverse kinematics and trajectory planning can be reduced to determining the spatial and temporal behavior of the curvature function, $\kappa = \kappa(s,t)$. That is, the function $\kappa(s,t)$ will determine the bending of the mechanism as a function of time, and is selected to achieve desired objectives, which depend upon the nature of the hyper-redundant manipulator application.

One method to make the kinematics problem tractable is to restrict the curvature function to a modal form:

$$\kappa(s,t) = \sum_{i=1}^{N} a_i(t)\phi_i(s) \qquad (5)$$

where the $\{\phi_i(s)\}$ are *mode functions*, the $\{a_i(t)\}$ are *modal participation factors*, and N is the number of chosen modes. Necessary conditions on the independence of the mode functions are detailed in [5]. In essence, this form of the curvature function restricts the effective degrees of freedom of the manipulator to the number needed to accomplish a task. Substituting (5) into (3) and (4), one sees that inverse kinematics and path planning of these complicated manipulators reduces to the search for the proper modal participation factors and their time varying behavior. A specific example is considered below. For additional details and examples of the modal approach to both planar and spatial hyper-redundant manipulator kinematics see [5,7].

1.3 A Closed Form Modal Solution

Consider the following choice of modes for $N = 2$,

$$\phi_1(s) = 2\pi \cos 2\pi s; \qquad \phi_2(s) = 2\pi \sin 2\pi s. \qquad (6)$$

Substituting these two modes into (3), (4), and (5), it can be shown using orthogonality properties and identities that the end-effector position ($s = 1$) is described by the following 'closed form' solution:

$$\begin{aligned} x_{ee} &= x_1(1) = \sin(a_2) J_0 \left[(a_1^2 + a_2^2)^{\frac{1}{2}}\right] \\ y_{ee} &= x_2(1) = \cos(a_2) J_0 \left[(a_1^2 + a_2^2)^{\frac{1}{2}}\right] \end{aligned} \qquad (7)$$

where $J_0(x)$ is the Bessel function of zero order. The inverse kinematics of this '2-participation factor' tensor manipulator can be computed with the following equations:

$$a_2 = \text{Atan2}(x_{ee}, y_{ee}) \qquad (8.a)$$

and

$$a_1 = \pm \left(\left[J_0^{-1}\left[(x_{ee}^2 + y_{ee}^2)^{\frac{1}{2}}\right]\right]^2 - [\text{Atan2}(x_{ee}, y_{ee})]^2 \right)^{\frac{1}{2}} \qquad (8.b)$$

The 'restricted inverse Bessel function of zero order', J_0^{-1} is defined as the inverse of the function $J_0(x)$ for $0 < x < \mu$ where $\mu \approx 3.832$ is the first local minimum of J_0, which is the first zero of J_1. In this way, the function is monotonically decreasing over the interval for which it is defined, and there is no problem in defining a unique inverse. If the inverse Bessel function were not restricted to this monotonic range, multiple solutions would be possible. The higher order solutions would physically correspond to configurations in which the manipulator self-intersected, and these solutions are neglected on the basis of practical considerations.
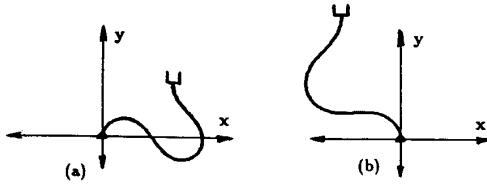

Figure 2: Two-Mode Inverse Kinematic Examples

Figures 2(a) and 2(b) show two examples of the inverse kinematic solutions provided by (8). In Figure 2(a), the manipulator end-effector is commanded to move to location $(x_{ee}, y_{ee}) = (0.3433, 0.2733)$. From (8), the modal participation factor solutions are $a_1 = 1.3588$ and $a_2 = 0.8984$. Figure 2(b) is another example with $(x_{ee}, y_{ee}) = (-0.2933, 0.6133)$, which results in modal participation factors of $a_1 = -1.0957$, $a_2 = -0.4461$.

Alternatives to closed form inverse kinematic solutions are resolved rate and "pseudo-dynamics" algorithms [7], which take the participation factors as generalized coordinates.

1.4. Overview of the Obstacle Avoidance Scheme

The method presented here deals with time-independent workspace environments. Furthermore, it is assumed that the layout of these obstacles in the workspace is well known, such as in an industrial setting, or assuming a sufficiently accurate vision sensing system. With these assumptions the problem of performing a task in a field of obstacles is equated to defining a path around obstacles to which the manipulator must adhere. Such a path, as illustrated in Figure (3), provides a trajectory or 'tunnel' in which the tensor-manipulator can 'slither' to circumvent the obstacles.

In practice, an automatic means for selecting one or more feasible tunnels which successfully negotiate the obstacle field could be generated using previously published methods. Free-space methods [3] based on generalized cones could be used to identify free path segments, and the centerlines of these segments could be assembled together to form the centerlines

of the "tunnels" required by this method. Similarly, cellular decomposition and contraction algorithms could be used to construct tunnel segments which avoid obstacles.

Once the end-effector has passed through the obstacle field, the portion of the manipulator in the region of the obstacles can be made to remain stationary. The unconstrainted portion of the manipulator can perform useful work. The modal approach to the kinematics of unconstrained tensor-manipulators can be used for the section of the manipulator outside the obstacle field. Essentially two problems must be overcome. First is how to determine the proper geometrical constraints on the manipulator to statically conform to the tunnel constraints. Second is how to account for the time rate of change of the manipulator curvature so that the moving manipulator obeys the geometric constraints of the tunnels as it moves through them. In this paper we do not assume or preclude any particular actuation scheme. However, we do assume that both the position and rate information are sufficient to drive the servo control algorithms of the mechanism actuators, which in turn controls the actual manipulator shape.

The next section presents the background mathematics and solutions to both the position and velocity kinematics problems.

2. Tunnel Constraints

In the position, or "shape" problem, we determine the curvature function of sections of the manipulator confined to tunnels. In addition, compatibility equations are developed for the free sections of the manipulator. In the velocity problem, the constraints are expressed in terms of allowable time behavior of the manipulator backbone curvature function. In this way the manipulator can slither through the tunnels from its starting configuration to its final configuration while obeying all of the geometric constraints.

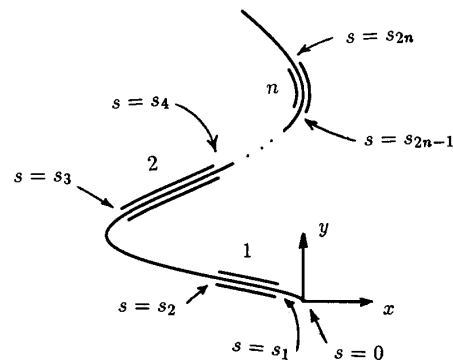### 2.1 Manipulator Shape with Tunnel Constraints

Figure 3: Tensor Manipulator Constrained to a System of Tunnels

For convenience, Equation (3) is written in complex notation

627

throughout this section. In other words, the position to every point on the manipulator is expressed as :

$$z(s,t) = x_1(s,t) + ix_2(s,t) = i \int_0^s e^{-i\theta(\sigma,t)}d\sigma. \qquad (9)$$

The orientation of every point is given by Equation (4).

Figure (3) shows a hyper-redundant manipulator in which certain segments of its length are constrained to pass through tunnel segments in order to avoid obstacles. Let the segments which are constrained to fit inside a tunnel be termed *interior* segments, while those segments which need not be constrained are termed *exterior* segments. Let us number the segments sequentially starting from the base of the manipulator. We will assume that the first segment is always an exterior segment, and therefore interior and exterior segments will respectively have even and odd indices.

For a tensor manipulator with some sections constrained to move in tunnels as shown in Figure (3), a curvature function which will satisfy the section by section constraints has the form:

$$\kappa(s,t) = \sum_{i=1}^{n} \kappa_i(s,t) W(s, s_{i-1}, s_i). \qquad (10)$$

where $s_0 = 0$ and $s_i = s_i(t)$ for $i > 0$, and i is an integer indexing a manipulator segment. The *window functions*, W, are defined as follows:

$$W(s, s_{i-1}, s_i) = H(s - s_i) - H(s - s_{i-1})$$

where H is the Heaviside step function. In other words, the curvature function is defined as a piecewise continuous function, where each segment of the manipulator is assigned a different local curvature function to satisfy its local constraints. For interior segments the curvature function, termed an *interior* curvature function, takes the form of a traveling wave:

$$\kappa_{2i} = \kappa_{2i}(s - s_{2i-1}(t))$$

whereas the curvature function describing exterior segments (an *exterior* curvature function) can have the more general form:

$$\kappa_{2i+1} = \kappa_{2i+1}(s,t).$$

However, the exterior segments have kinematic restrictions on the position and orientation of the manipulator at the entrance and exit of interior segments:

$$\int_{s_{2i}}^{s_{2i+1}} \kappa_{2i+1}ds = \theta_{2i}^{2i+1} = const_1 \qquad (11)$$

and

$$i \int_{s_{2i}}^{s_{2i+1}} \exp\left[-i\int_{s_{2i}}^{s} \kappa_{2i+1}d\sigma\right] ds = z_{2i}^{2i+1} = const_2. \qquad (12)$$

θ_{2i}^{2i+1} and z_{2i}^{2i+1} are the orientation and position of the frame at the entrance of the i+1st tunnel with respect to the frame at the exit of the ith tunnel.

In this way, the manipulator backbone curve is at least a once continuously differentiable curve along its whole length.

2.2 Manipulator Velocities with Tunnel Constraints

This subsection addresses the computationally more difficult problem of what the instantaneous velocities are at every point along the tensor manipulator. Abstractly, all that needs to be done is to take the time derivative of position to get the velocities. However, some mathematical manipulation can be performed so that both physical insight and simplification of the computational algorithm needed to implement these equations can be achieved.

The velocity constraints on the free sections of the manipulator are found by simply taking the time derivative of the position constraints represented by (11) and (12) to yield:

$$\dot{\theta}_{2i}^{2i+1} = \frac{d}{dt}\int_{s_{2i}}^{s_{2i+1}} \kappa_{2i+1}ds = 0 \qquad (13)$$

and

$$\dot{z}_{2i}^{2i+1} = i\frac{d}{dt}\int_{s_{2i}}^{s_{2i+1}} \exp\left[-i\int_{s_{2i}}^{s} \kappa_{2i+1}d\sigma\right] ds = 0. \qquad (14)$$

Knowing that $s_k = s_k(t)$, Liebnitz's rule can be used to write the velocity constraints (13) and (14) explicitly as:

$$\int_{s_{2i}}^{s_{2i+1}} \frac{\partial\kappa_{2i+1}}{\partial t}ds + \kappa_{2i+1}(s_{2i+1},t)\dot{s}_{2i+1} \\ - \kappa_{2i+1}(s_{2i},t)\dot{s}_{2i} = 0 \qquad (15)$$

and

$$-i\int_{s_{2i}}^{s_{2i+1}} \frac{\partial\theta_{2i}}{\partial t}e^{-i\theta_{2i}}ds + e^{-i\theta_{2i}(s_{2i+1},t)}\dot{s}_{2i+1} - \dot{s}_{2i} = 0. \qquad (16)$$

where

$$\theta_{2i}(s,t) = \int_{s_{2i}}^{s} \kappa(\sigma,t)d\sigma.$$

Note that for the parts of the manipulator which fully occupy a tunnel, $s_{2i+1} - s_{2i} = const$, which can also be written

$$\dot{s}_{2i+1} = \dot{s}_{2i}.$$

The velocity of every point along the manipulator is of the form:

$$v(s,t) = i\frac{\partial}{\partial t}\int_0^s e^{-i\theta(\sigma,t)}d\sigma = \int_0^s \frac{\partial\theta}{\partial t}e^{-i\theta}d\sigma. \qquad (17)$$

Using the properties of generalized distributions, ie, window, step, and delta functions, it can be shown that

$$\frac{\partial\theta}{\partial t}(s,t) = \sum_{i=1}^{n} \left[\int_{s_{i-1}}^{s_i} \frac{\partial\kappa_i}{\partial t}(\sigma,t)d\sigma + \dot{s}_i j(s_i,t)H(s - s_i)\right] \qquad (18)$$

where

$$j(s_i,t) = \kappa_i(s_i^-,t) - \kappa_{i+1}(s_i^+,t)$$

is the 'jump' in the curvature between adjacent sections.

Substituting (18) into (17) provides an expression which can be evaluated numerically, to find the velocity of every point of the manipulator at any instant in time. The next section uses the methods developed here in examples to illustrate how position and velocity of all points on the backbone curve

confined to a system of tunnels can be defined using the above formulation.

2.3. A Specific Example

Figures (4) and (5) illustrate a specific example of the general formulation of the previous subsection. The hyperredundant manipulator must pass through a single maze-like tunnel and reach a goal on the other side. In the first unconstrained exterior section, i.e., $0 < s < s_1$, the curvature function will be assumed to have the form

$$\kappa_1(s,t) = a(t)\cos\frac{2\pi s}{s_1(t)}$$

so that a point, $\overline{x}(s_1)$, on the moving (or "slithering") manipulator is always coincident with the tunnel entrance.

Integrating the forward kinematics equations (3), or (9), we get

$$x_1(s_1) = \int_0^{s_1} \sin(\frac{as_1}{2\pi}\sin\frac{2\pi s}{s_1})ds = 0 \qquad (19)$$

and

$$x_2(s_1) = \int_0^{s_1} \cos(\frac{as_1}{2\pi}\sin\frac{2\pi s}{s_1})ds = J_0(\frac{as_1}{2\pi})s_1. \qquad (20)$$

Let $h > 0$ be the distance from the base of the manipulator to the entrance of the tunnel. h is a constant and the following must hold:

$$h = J_0\left(\frac{as_1}{2\pi}\right)s_1. \qquad (21)$$

For h to be positive, it is necessary that:

$$0 < \frac{as_1}{2\pi} < \mu_1 \qquad (22)$$

where $\mu_1 \approx 2.405$ is the first zero of J_0.

This being defined, the condition for stationarity of $\overline{x}(s_1)$ at the entrance of the tunnel, while s_1 changes, is

$$a(t) = \frac{2\pi}{s_1(t)}J_0^{-1}\left(\frac{h}{s_1(t)}\right) \qquad (23)$$

and so

$$\kappa_1(s,t) = \frac{2\pi}{s_1(t)}J_0^{-1}\left(\frac{h}{s_1(t)}\right)\cos\frac{2\pi s}{s_1(t)}. \qquad (24)$$

The obstacle environment is illustrated in Figure 4. With $\overline{x}(s_1) = [0,h]^T$ fixed, we can now determine an appropriate $\kappa_2(s,t)$ which is the curvature of the section of manipulator confined to the tunnel. For this particular obstacle field κ_2 can be defined as follows:

$$\kappa_2(s,t) = -\frac{\pi}{L}[W(s,s_1,s_1+L) - W(s,s_1+L,s_1+2L)$$
$$+W(s,s_1+2L,s_1+3L)]. \qquad (25)$$

The composite curvature function for this example is then

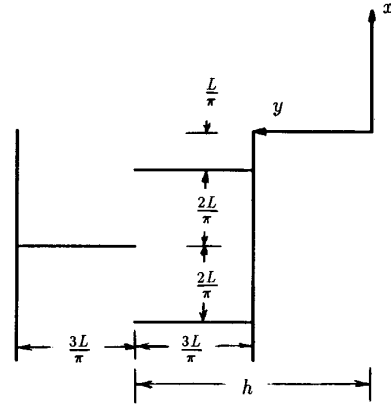$$\kappa(s,t) = \kappa_1(s,t)W(s,0,s_1) + \kappa_2(s,t)W(s,s_1,1) \qquad (26)$$

Figure 4: Dimensions of the Obstacle Field

The given choice of κ_2 corresponds to three consecutive semicircular arcs. The magnitude of the curvature over each of the three sections, each with arc length L, is π/L, corresponding to semicircles of radius $r = L/\pi$. The curvature function of circular arcs is always the reciprocal of the radius of the circle to which the arcs belong. The window functions take the value of unity over each of the semicircles, and the sign indicates the sense in which the arc turns. A positive sign indicates clockwise bending, and a negative sign indicates counterclockwise bending of the manipulator.

Integrating Equation (26) in the variable s, as in Equation (4), we find that

$$\theta(s,t) = J_0^{-1}\left(\frac{h}{s_1}\right)\sin\left(\frac{2\pi s}{s_1}\right) - \frac{\pi}{L}(s-s_1)W(s,s_1,s_1+L)$$
$$+ [\frac{\pi}{L}(s-s_1-L)-\pi]W(s,s_1+L,s_1+2L)$$
$$- \frac{\pi}{L}(s-s_1-2L)W(s,s_1+2L,s_1+3L)$$
$$- \pi W(s,s_1+3L,1). \qquad (27)$$

A time history of this tunneling obstacle avoidance maneuver is shown in Figure 5, corresponding to the obstacle of Figure 4. The configurations shown correspond to $h = 0.4$ and $s_1 = 1.00, 0.94, 0.83, 0.68, 0.54, 0.44$. The dimensions of the obstacle field are shown in Figure 4, where in the numerical example presented, $L = 0.2$.

These results can be used to assemble obstacle avoidance curvature functions for arbitrary obstacle fields. If a feasible obstacle avoidance path of finite width exists in an obstacle field, one can always define a tunnel of suitable accuracy for avoiding obstacles as a sequence of straight lines and circular arcs segments which are differentiable at their junction. Consequently, the results above and in [5] can be used with a free space algorithm to develop a general planar obstacle avoidance scheme for this class of manipulators.
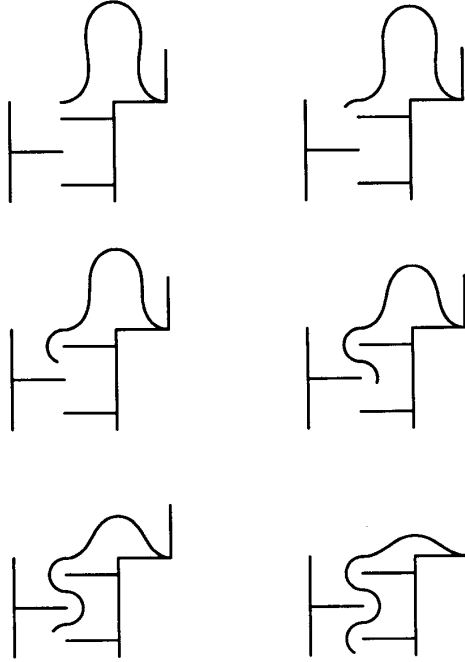
Figure 5: Continuous Hyper-Redundant Manipulator in an Obstacle Field

3. Fitting the Continuous Solution to a Discrete Manipulator

While the modal method has been developed under the assumption that the backbone curve of the tensor manipulator is continuous and can conform to any given shape, a discrete-linked manipulator is much more restricted in its physical capabilities. This section considers the adaptation of the current analysis to the case of an n-link planar revolute manipulator, and is a summary of results in [5]. In this shape fitting method, the discrete manipulator geometry is selected so that the nominal shape of the discrete manipulator coincides as closely as possible with the continuous backbone curve solution.

Figure 1(a) shows a planar manipulator comprised of n rigid links with n revolute joints. All links are assumed to be of the same length, and the total length of the manipulator is normalized to 1. The forward kinematics for this manipulator is:

$$x_{ee} = \frac{1}{n} \sum_{i=1}^{n} \sin \left(\sum_{j=1}^{i} \theta_j \right) \tag{28}$$

$$y_{ee} = \frac{1}{n} \sum_{i=1}^{n} \cos \left(\sum_{j=1}^{i} \theta_j \right) \tag{29}$$

$$\theta_{ee} = \sum_{i=1}^{n} \theta_i \tag{30}$$

where x_{ee}, y_{ee}, and θ_{ee} denote the position and orientation of the end effector.

The discrete manipulator shape and end-effector location are fitted to the continuous curve model by minimizing the sum of squared distance between points on both manipulators located at $s = \frac{i}{n}$ for $i = 1, ..., n$. In this method, the following is assumed:

$$\theta_{j+1} = \theta \left(\frac{2j+1}{2n} \right) - \theta \left(\frac{2j-1}{2n} \right) + \epsilon_j. \tag{31}$$

where $\theta(\alpha) = 0$ for all $\alpha < 0$. In other words, we take the angles of the discrete case to be approximately the change in angle over a corresponding section of length in the continuous case. Note that there are n free 'fitting' parameters, ϵ_j, This extra freedom can be used to make the discrete link model conform even further to the continuous case. Below is an expression of the squared distance from the joint points on the discrete manipulator to the corresponding points on the continuous manipulator, i.e, the points $s = \frac{i}{n}$. The function to be minimized is the sum of the contributions from all such points.

$$G = \frac{1}{2} \sum_{k=1}^{n} \left[\left(\int_0^{\frac{k}{n}} \sin \theta ds - \frac{1}{n} \sum_{i=1}^{k} \sin \sum_{j=1}^{i} \theta_j \right)^2 + \left(\int_0^{\frac{k}{n}} \cos \theta ds - \frac{1}{n} \sum_{i=1}^{k} \cos \sum_{j=1}^{i} \theta_j \right)^2 \right] \tag{32}$$

Linearizing the above expression while assuming ϵ_j are small in (31), and using the definitions:

$$x_k = \int_0^{\frac{k}{n}} \sin \theta ds \qquad y_k = \int_0^{\frac{k}{n}} \cos \theta ds$$

$$s_{r,k} = \frac{1}{n} \sum_{i=r}^{k} \sin \left(\theta \left(\frac{2i+1}{2n} \right) \right)$$

$$c_{r,k} = \frac{1}{n} \sum_{i=r}^{k} \cos \left(\theta \left(\frac{2i+1}{2n} \right) \right)$$

the explicit form of the mth component of the linearized optimization of least squares distance is:

$$\sum_{p=m}^{n} \left[\left(x_p - s_{1,p} - \sum_{i=1}^{p} \epsilon_i c_{i,p} \right) c_{m,p} - \left(y_p - c_{1,p} + \sum_{i=1}^{p} \epsilon_i s_{i,p} \right) s_{m,p} \right] = 0 \tag{33}$$

providing n linear equations to specify the n variable ϵ_i's.

If the $\{\epsilon_i\}$ are calculated, and they are small, then the assumptions allowing linearization are justified. If however, the values of ϵ_i are not sufficiently small for the assumptions to be valid, then the linear approximation fitting procedure can be iterated, or a much less efficient nonlinear solution procedure must be implemented. This will generally only be

the case when there are points on the backbone curve with large curvature and/or the discrete manipulator has a small number of links. Figure 6 shows an example of a 10 link manipulator which has been 'fitted' to the configurations in Figure (5), demonstrating a case where this method works quite well.
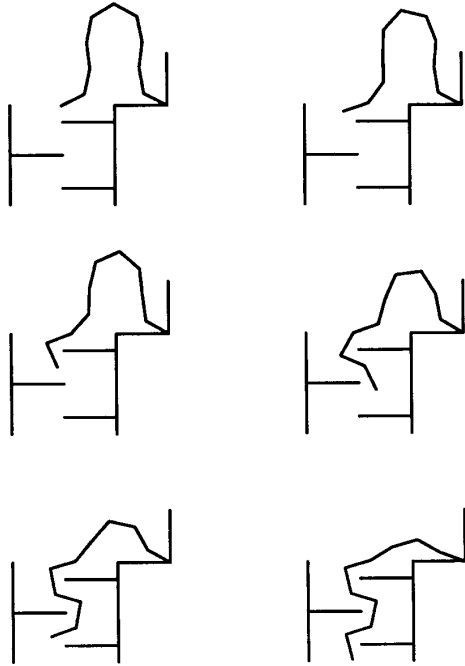

Figure 6: Discrete Hyper-Redundant Manipulator Obstacle Avoidance

4. Conclusions

This report presented a novel obstacle avoidance concept, based on "tunneling," for hyper-redundant manipulators of constant length. A general formulation was given which allows a manipulator to maneuver through a complicated sequence of interior and exterior segments. Computer simulations were presented for a particular example. In principle, connecting circular arcs and line segments as in the presented example can be used to construct a system of tunnels for maneuvering in arbitrarily complex obstacle fields. The benefit of this method over existing potential field and optimal methods for application to hyper-redundant manipulators is that a comparatively efficient set of kinematic equations based on differential geometry need be computed, thus allowing much faster solutions. The method was demonstrated for both continuous and discrete hyper-redundant manipulators. While the applicability of the present model diminishes with diminishing degree of redundancy, it works quit well in situations such as the continuous manipulator case, where conventional methods of analysis do not apply.

References

[1] J. Baillieul, "Avoiding Obstacles and Resolving Kinematic Redundancy," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, 1986, pp. 1698-1704.

[2] Bowman, F., Introduction to Bessel Functions, Dover Inc, New York, 1958.

[3] R.A. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, March 1983, pp. 190-197.

[4] J. Canny, The Complexity of Robot Motion Planning, The MIT Press, Cambridge, MA, 1988.

[5] Chirikjian, G.S., Burdick, J.W., "A Modal Approach to the Kinematics of Hyper-Redundant Manipulators," *Robotics and Mechanical Systems Report* No. RMS-89-3, Dept. of Mechanical Engineering, California Institute of Technology, Pasadena, CA, 91125, September 1989.

[6] Chirikjian, G.S., Burdick, J.W., "An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators," *Robotics and Mechanical Systems Technical Report* No. RMS-89-4, Dept. of Mechanical Engineering, California Institute of Technology, Sept. 1989.

[7] Chirikjian, G.S., Burdick, J.W., "Pseudo-Dynamics Algorithms for Computing Inverse Kinematics and Path Planning of Robotic Manipulators," *Robotics and Mechanical Systems Technical Report* No. RMS-89-5, Dept. of Mechanical Engineering, California Institute of Technology, March 1990.

[8] Chirikjian, G.S., Burdick, J.W., "Hyper-Redundant Robotic Locomotion: Locomotion Without Wheels Tracks, or Legs," *Robotics and Mechanical Systems Technical Report* No. RMS-89-6, Dept. of Mechanical Engineering, California Institute of Technology, March 1990.

[9] Hirose, S., Umetani, Y., "Kinematic Control of Active Cord MechanismWith Tactile Sensors," *Proceedings of Second International CISM-IFT Symposium on Theory and Practice of Robots and Manipulators*, pp. 241-252, 1976.

[10] Hemami, A.,"Studies on a Light Weight and Flexible Robot Manipulator," *Robotics*, Vol. 1, 1985, pp.27-36.

[11] Ivanescu, M., Badea, I.,"Dynamic Control for a Tentacle Manipulator," Int. Conf. on Robotics and Factories of the Future, pp. 317-328, Dec 4-7, 1984, Charlotte, Nc, USA.

[12] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. J. Robotics Research, Vol. 5, No. 1, 1986.

[13] P. Khosla and R. Volpe, "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, 1988, pp. 1778-1784.

[14] C.A. Klein and C.H. Huang, "Review of the Pseudoinverse for Control of Kinematically Redundant Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, March 1983.

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

[16] Shahinpoor, M., Kalhor, H., Jamshidi, M., "On Magnetically Activated Robotic Tensor Arms," Proceedings of the International Symposium on Robot Manipulators: Modeling, Control, and Education, Nov. 12-14, 1986, Albuquerque, New Mexico.

[17] Tesar, D., Butler, M.S., "A Generalized Modular Architecture for Robot Structures," ASME Manufacturing Review, Vol. 2, No. 2, June 1989, pp. 91-118.