Workspace Generation of Hyper-Redundant Manipulators as a Diffusion Process on $SE(N)$

Yunfeng Wang, *Member, IEEE,* and Gregory S. Chirikjian, *Member, IEEE*

***Abstract*—Hyper-redundant manipulators have a large number of redundant degrees of freedom. They have been recognized as a means to improve manipulator performance in complex and unstructured environments. However, the high degree of redundancy also causes difficulty in the calculation of workspaces and inverse kinematics. This paper develops a diffusion-based algorithm for workspace generation of hyper-redundant manipulators. This algorithm makes the workspace generation problem as simple as solving a diffusion equation which has an explicit solution. This diffusion equation is a partial differential equation defined on the motion group $SE(N)$, and describes the evolution of the workspace density function, depending on manipulator length and kinematic properties. This paper also solves the inverse kinematics problem in an elegant way by dividing the manipulator into virtual segments and cascading the corresponding workspace densities generated by the diffusion equation.**

***Index Terms*—Diffusion process, Euclidean motion group, harmonic analysis, hyper-redundant manipulator, inverse kinematics, workspace generation.**

NOMENCLATURE

$SE(N)$	N-dimensional Euclidean motion group.
I	Identity matrix.
$\mathcal{J}_L$	Left Jacobian.
$\mathcal{J}_R$	Right Jacobian.
$\tilde{X}_i$	Basis element of the Lie algebra for $SE(2)$.
$\tilde{X}_i^R$	Right differential operator corresponding to $\tilde{X}_i$.
$\tilde{X}_i^L$	Left differential operator corresponding to $\tilde{X}_i$.
$f(g;L)$	Workspace density for a manipulator of length L. Here $g \in SE(2)$.
$\hat{f}(p)$	Motion group Fourier transform of $f(g)$.
$\mathcal{F}(f)$	Motion group Fourier transform of $f(g)$.
$\eta(\tilde{X}_i, p)$	Operational property coefficient matrix for $\tilde{X}_i$.
η_{mn}	$m-n$ element of the operational property coefficient matrix η.
$\alpha, \beta, \varepsilon$	Kinematic parameters for the diffusion equation.

Manuscript received April 2, 2003. This paper was recommended for publication by Associate Editor G. Oriolo and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant IIS-0098382. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Washington, DC, May 11–15, 2002.

Y. F. Wang is with the Department of Engineering, The College of New Jersey, Ewing, NJ 08628 USA (e-mail: jwang@tcnj.edu).

G. S. Chirikjian is with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: gregc@jhu.edu).

Digital Object Identifier 10.1109/TRA.2004.825473

I. INTRODUCTION

HYPER-REDUNDANT manipulators, which are also called snake-like, serpentine, or highly articulated manipulators, possess conformational freedom far superior to that of conventional manipulators. They have great potential for applications where a high degree of redundancy is essential. Examples include inspection and repair tasks in complex environments, search and rescue tasks in areas difficult to access by humans, and medical diagnostic and minimally invasive operations in health care.

Substantial research has been done on hyper-redundant manipulators, and many different prototypes have been constructed. Cable/tendon-driven mechanisms have been used in [12], [14], and [20]. Elastic links are employed in [7], [13], and [15]. The design of the Jet Propulsion Laboratory (JPL, Pasadena, CA) serpentine manipulator [26] uses gears with a high gear ratio. The manipulator in [16] is composed of serially connected three-bar linkages. To avoid complicated control schemes, binary actuated manipulators are proposed in [5]. They adopt a variable geometry truss structure, and are actuated by pneumatic cylinders.

The algorithmic issues associated with hyper-redundant manipulators such as kinematics [11], control [22], [24], [32], and motion planning [6], [21] also have been explored extensively. In this paper, we concentrate on the workspace generation problem for hyper-redundant manipulators. Geometric methods were used in [18] to generate the workspace for a manipulator with an arbitrary number of revolute joints. A curve-approximation approach was presented to determine the workspace of complex planar manipulators in [28]. The authors of [27] divided the manipulator into parts, subworkspaces of which were then calculated using the Jacobian. The Monte Carlo method was used in [1], where a large number of random actuator values were generated, and the corresponding reachable positions were calculated. In terms of workspace density, the author of [9] presented a method based on concatenation of the densities of individual modules by sweeping, while the authors of [4] applied the convolution of functions on Lie groups to determine the workspaces through partitioning a manipulator into segments, and approximating the workspace of each segment as a density function.

Different from all the above methods, we formulate the workspace generation problem as a diffusion process and develop a diffusion-based algorithm. In this algorithm, the workspace density is the solution to a partial differential equation defined on the motion group $SE(N)$. This partial differential equation describes the evolution of the workspace

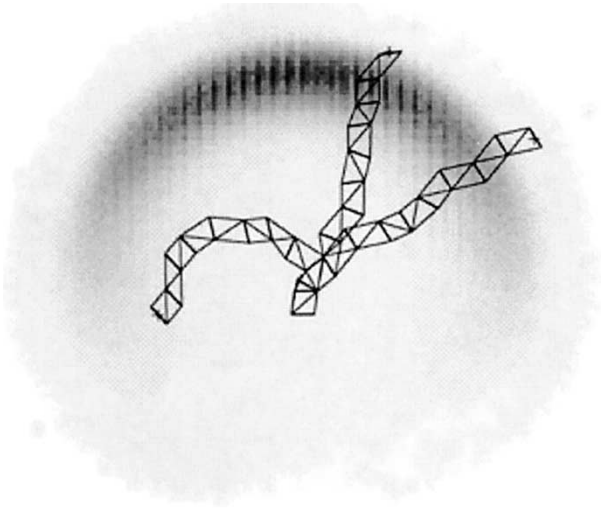

Fig. 1. Workspace density of a binary manipulator with three manipulator configurations displayed [4].

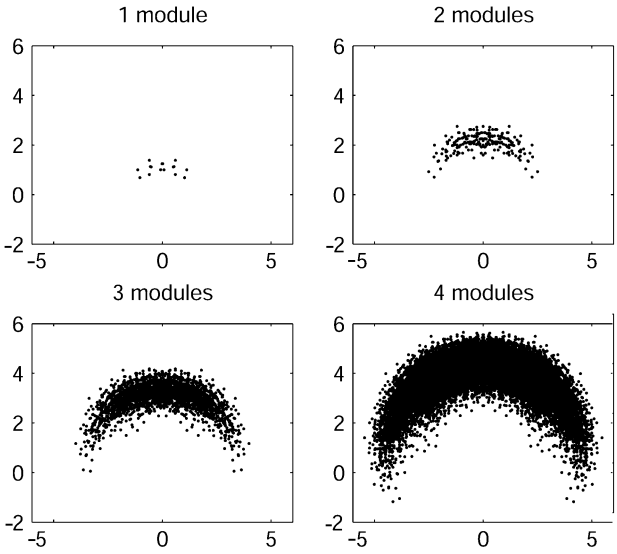

Fig. 2. Workspace of a manipulator with different numbers of modules.

density function, depending on manipulator length and kinematic properties, and can be easily solved using the techniques of noncommutative harmonic analysis as developed in [2].

II. Inspiration of the Algorithm

A. Workspace Density

Our diffusion-based algorithm takes advantage of the concept of workspace density. This concept was first proposed in [29] to analyze the accuracy of performance of a manipulator by discretizing the range of continuously actuated joints. Later, this concept was employed to handle workspace generation and kinematics problems for binary manipulators in [4] and [9].

Workspace density describes the density of reachable points/frames in any portion of the workspace [4], where the workspace is discretized into small blocks, and the density of points/frames is the number of reachable points/frames per unit workspace volume. Workspace density is a probabilistic measure of accuracy over the workspace. The density in the neighborhood of a given point/frame is an indication of how accurately a discretely actuated manipulator can reach that point/frame. Higher density of a point means the manipulator can reach that point more accurately. The workspace density function is a probability density function (pdf) that describes the distribution of points/frames over the workspace.

When the concept of workspace density is used for continuously actuated manipulators, the range of motion of each degree of freedom (DOF) will be discretized first. For example, a manipulator with n DOFs, each sampled at K values, results in a discretized workspace with K^n positions and orientations.

Fig. 1 displays the workspace density of a planar binary manipulator with ten modules [4]. Three of the manipulator's 2^{30} configurations are indicated. Darker areas mean higher density, and the manipulator can reach that part more accurately.

B. Physical Analogy

Consider a discretely actuated serial manipulator where each module can reach 16 different states. Such modules can be designed by using binary actuators as the vertical elements in each truss, and two stacked binary actuators of different stroke lengths as the diagonal element. The workspaces of this manipulator with different modules are shown in Fig. 2. They are generated by brute-force enumeration. It is easy to notice that the size of the workspace spreads out as the number of modules increases. This enlargement of the workspace is just like the diffusion produced by a drop of ink spreading in a cup of water. Inspired by this observation, we view the workspace of a hyper-redundant manipulator as something that grows/evolves from a single point source at the base as the length of the manipulator increases from zero. The workspace is then generated after the manipulator grows to full length.

In this way, the workspace generation problem is described by a diffusion process. Is this intuitive analogy valid? It has been shown that the workspace density of any discrete-state serial manipulator (or any manipulator whose joint ranges are sampled at discrete values) can be generated by dividing the manipulator into pieces, exhaustively enumerating the workspace density of each piece, and then generating the workspace density for the whole manipulator by computing the $SE(N)$-convolution for adjacent pieces [4], [9], [19]. This fact is a consequence of the Markovian nature of the workspace density. The well-established theory of Markov processes associates with every continuous Markov process a diffusion equation that describes the evolution of probability density. That is, given a Chapman–Kolmogorov (i.e., convolution) equation, there is an underlying diffusion process which can be described by a Fokker–Planck equation.

With the knowledge gained from our study on the conformational statistics of stiff macromolecules [3] and the results of [19], we derive a diffusion equation describing the evolution of the workspace density function depending on manipulator length and kinematic properties. This equation is a partial differential equation defined on the motion group, and can be easily solved using the the techniques of noncommutative harmonic analysis.

III. Noncommutative Harmonic Analysis

Noncommutative harmonic analysis is a generalization of Fourier analysis. This mathematical tool was developed for the case of rigid-body motions by and for pure mathematicians and theoretical physicists in the 1960's [23]. It is rarely known by engineering scholars. In this paper, it is successfully applied to formulate and solve problems in the field of robotics. This section reviews this powerful mathematical tool and some new results we derived.

A. Euclidean Motion Group

The Euclidean motion group, $SE(N)$, is the semidirect product of $\mathbb{R}^N$ with the special orthogonal group, $SO(N)$.[1] We denote elements of $SE(N)$ as $g = (A, \mathbf{a})$ where $A \in SO(N)$ and $\mathbf{a} \in \mathbb{R}^N$. For any $g = (A, \mathbf{a})$ and $h = (B, \mathbf{b}) \in SE(N)$, the group law and inverse of an element are, respectively, written as $g \circ h = (AB, \mathbf{a} + A\mathbf{b})$ and $g^{-1} = (A^T, -A^T\mathbf{a})$. It is often convenient to express an element of $SE(N)$ as an $(N+1) \times (N+1)$ homogeneous transformation matrix of the form

$$g = \begin{pmatrix} A & \mathbf{a} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

In this way, rotation and translation are combined into a single matrix. A homogeneous transformation matrix takes the place of the pair $(A, \mathbf{a})$, and the group operation becomes the matrix multiplication.

For example, each element of $SE(2)$ parameterized using polar coordinates can be written as

$$g(\phi, r, \theta) = \begin{pmatrix} \cos\phi & -\sin\phi & r\cos\theta \\ \sin\phi & \cos\phi & r\sin\theta \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where $0 \leq \phi, \theta \leq 2\pi$, and $0 \leq r \leq \infty$. $SE(2)$ is a three-dimensional (3-D) manifold much like $\mathbb{R}^3$. We can integrate over $SE(2)$ using the volume element $d(g(\phi, r, \theta)) = (1/4\pi^2) r dr d\theta d\phi$ [2].

Our diffusion model for workspace generation of planar manipulators is defined on the motion group $SE(2)$. We focus on $SE(2)$ because the mathematics is less involved, and the methodology is the same as it would be in the 3-D case. The interested reader is referred to [2] for the mathematical background required to apply this methodology to spatial manipulators.

B. Jacobians for the Motion Group $SE(2)$

Jacobian matrices are important for understanding differential operators and integration defined on motion groups. They can be derived by equating two different looking, though equivalent, ways of writing $\dot{g}g^{-1}$ and $g^{-1}\dot{g}$ for $g \in SE(2)$ where $\dot{g} = dg/dt$. One way is to use matrix multiplication and the other is to use the chain rule from calculus. When $\dot{g}$ is on the left or right side of g^{-1}, we call the corresponding Jacobian a left or right Jacobian.

[1]The group $SO(N)$ consists of $N \times N$ matrices with the properties $RR^T = I$ and $\det R = +1$. The group law is matrix multiplication.

Let $g \in SE(2)$ be parameterized with (q_1, q_2, q_3) as

$$g(\mathbf{q}) = \begin{pmatrix} A(\mathbf{q}) & \mathbf{a}(\mathbf{q}) \\ \mathbf{0}^T & 1 \end{pmatrix}$$

where (q_1, q_2, q_3) is written as a vector $\mathbf{q} \in \mathbb{R}^3$. Using matrix multiplication, we have

$$\dot{g}g^{-1} = \begin{pmatrix} \dot{A}A^T & -\dot{A}A^T\mathbf{a} + \dot{\mathbf{a}} \\ \mathbf{0}^T & 0 \end{pmatrix}. \tag{2}$$

It is well known that $\dot{A}A^T$ is a 2-D skew-symmetric matrix [8] with the form

$$\Omega = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}.$$

We denote $\mathrm{vect}(\Omega) = \omega \in \mathbb{R}$. Following [25], we define an operation $()^\vee$ such that

$$\begin{pmatrix} \Omega & \mathbf{v} \\ \mathbf{0}^T & 0 \end{pmatrix}^\vee = \begin{pmatrix} \omega \\ \mathbf{v} \end{pmatrix}.$$

Applying this operation to (2), we get

$$(\dot{g}g^{-1})^\vee = \begin{pmatrix} \omega_L \\ \mathbf{v}_L \end{pmatrix} \tag{3}$$

where $\omega_L = \mathrm{vect}(\dot{A}A^T)$, and $\mathbf{v}_L = -\dot{A}A^T\mathbf{a} + \dot{\mathbf{a}}$. Using the chain rule, we find

$$\dot{g}g^{-1} = \left[\frac{\partial g}{\partial q_1}g^{-1}, \frac{\partial g}{\partial q_2}g^{-1}, \frac{\partial g}{\partial q_3}g^{-1}\right] \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}. \tag{4}$$

From (3) and (4), we have

$$\begin{pmatrix} \boldsymbol{\omega}_L \\ \mathbf{v}_L \end{pmatrix} = \mathcal{J}_L(\mathbf{q})\dot{\mathbf{q}}$$

where

$$\mathcal{J}_L(\mathbf{q}) = \left[\left(\frac{\partial g}{\partial q_1}g^{-1}\right)^\vee, \left(\frac{\partial g}{\partial q_2}g^{-1}\right)^\vee, \left(\frac{\partial g}{\partial q_3}g^{-1}\right)^\vee\right] \tag{5}$$

is the left Jacobian. Similarly, we can get the right Jacobian as

$$\mathcal{J}_R(\mathbf{q}) = \left[\left(g^{-1}\frac{\partial g}{\partial q_1}\right)^\vee, \left(g^{-1}\frac{\partial g}{\partial q_2}\right)^\vee, \left(g^{-1}\frac{\partial g}{\partial q_3}\right)^\vee\right]. \tag{6}$$

For example, when $g \in SE(2)$ is parameterized as in (1), we can use (5) to get the corresponding left Jacobian as

$$\mathcal{J}_L = \begin{pmatrix} 1 & 0 & 0 \\ r\sin\theta & \cos\theta & -r\sin\theta \\ -r\cos\theta & \sin\theta & r\cos\theta \end{pmatrix} \tag{7}$$

and (6) to get the corresponding right Jacobian as

$$\mathcal{J}_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi-\theta) & r\sin(\phi-\theta) \\ 0 & -\sin(\phi-\theta) & r\cos(\phi-\theta) \end{pmatrix}. \tag{8}$$

The bi-invariant integration measure for $SE(2)$ is defined to within an arbitrary constant as $|\det J_R| dr d\phi d\theta = |\det J_L| dr d\phi d\theta$.

C. Differential Operators for the Motion Group $SE(2)$

We take a very coordinate-dependent approach to describe these concepts. For readers interested in coordinate-free approaches, see [2], [25], and references therein.

For small translational (rotational) displacements from the identity along (about) the ith coordinate axis, the homogeneous

transforms representing infinitesimal motions can be expressed by

$$\exp(t\tilde{X}_i) \approx I + t\tilde{X}_i \; (t \ll 1)$$

where

$$\tilde{X}_1 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$\tilde{X}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$\tilde{X}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

$\tilde{X}_1, \tilde{X}_2$, and $\tilde{X}_3$ correspond, respectively, to infinitesimal rotation about the Z axis and translations along the X and Y axes. They are the natural basis elements of the Lie algebra for $SE(2)$. It is often convenient to write them in vector form as

$$(\tilde{X}_1)^\vee = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad (\tilde{X}_2)^\vee = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad (\tilde{X}_3)^\vee = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Let $g \in SE(2)$ and $f(g)$ be a function which takes elements of the motion group as its arguments. Analogous to the partial derivative of a function of an $\mathbb{R}^N$-valued argument, we can define differential operators acting on functions of the motion group as [10], [23]

$$\tilde{X}_i^L f \triangleq \frac{d}{dt} f(\exp(-t\tilde{X}_i) \circ g)\,|_{t=0}$$
$$\tilde{X}_i^R f \triangleq \frac{d}{dt} f(g \circ \exp(t\tilde{X}_i))\,|_{t=0}.$$

In our notation, the superscripts L and R denote whether the infinitesimal motion $\exp(t\tilde{X}_i)$ is on the left or the right of g. Hence, $\tilde{X}_i^L$ is invariant under right shifts, and $\tilde{X}_i^R$ is invariant under left shifts.

Let us define the parameters $\mathbf{q}^{R,i}(t)$ such that

$$g(\mathbf{q}) \circ (I + t\tilde{X}_i) = g(\mathbf{q}^{R,i}). \tag{9}$$

Performing the matrix multiplication on the left and expanding the right side of (9) in a Taylor series to first order in t, we get

$$g + tg\tilde{X}_i = g + t \sum_{j=1}^{3} \frac{\partial g}{\partial q_j} \frac{dq_j^{R,i}}{dt}\bigg|_{t=0}. \tag{10}$$

We then have that

$$\tilde{X}_i = \sum_{j=1}^{3} g^{-1} \frac{\partial g}{\partial q_j} \frac{dq_j^{R,i}}{dt}\bigg|_{t=0}$$

or

$$(\tilde{X}_i)^\vee = \sum_{j=1}^{3} \left(g^{-1} \frac{\partial g}{\partial q_j} \right)^\vee \frac{dq_j^{R,i}}{dt}\bigg|_{t=0}.$$

Using the right Jacobian for the motion group as given in (6), we can rewrite

$$(\tilde{X}_i)^\vee = \mathcal{J}_R(\mathbf{q}) \frac{d\mathbf{q}^{R,i}}{dt}\bigg|_{t=0}.$$

This allows us to solve for

$$\frac{d\mathbf{q}^{R,i}}{dt}\bigg|_{t=0} = \mathcal{J}_R^{-1}(\tilde{X}_i)^\vee$$

which is used to calculate the right differential operators for $SE(2)$ as

$$\tilde{X}_i^R f = \mathcal{J}_R^{-1} \sum_{j=1}^{3} (\tilde{X}_i)^\vee \cdot \mathbf{e}_j \frac{\partial f}{\partial q_j} = \left(\mathcal{J}_R^{-1}\mathbf{e}_i\right) \cdot (\nabla_{\mathbf{q}} f) \tag{11}$$

where $\mathbf{e}_1 = [1,0,0]^T, \mathbf{e}_2 = [0,1,0]^T$, and $\mathbf{e}_3 = [0,0,1]^T$. Similarly, we can derive the left differential operators for $SE(2)$ as

$$\tilde{X}_i^L f = -\mathcal{J}_L^{-1} \sum_{j=1}^{3} (\tilde{X}_i)^\vee \cdot \mathbf{e}_j \frac{\partial f}{\partial q_j} = -\left(\mathcal{J}_L^{-1}\mathbf{e}_i\right) \cdot (\nabla_{\mathbf{q}} f). \tag{12}$$

Substituting (8) into (11), we can get explicit expressions of the right differential operators $\tilde{X}_i^R$ for $SE(2)$ in polar coordinates as

$$\tilde{X}_1^R = \frac{\partial}{\partial \phi}$$
$$\tilde{X}_2^R = \cos(\phi - \theta)\frac{\partial}{\partial r} + \frac{\sin(\phi - \theta)}{r}\frac{\partial}{\partial \theta}$$
$$\tilde{X}_3^R = -\sin(\phi - \theta)\frac{\partial}{\partial r} + \frac{\cos(\phi - \theta)}{r}\frac{\partial}{\partial \theta}.$$

Substituting (7) into (12), we can get explicit expressions of the left differential operators $\tilde{X}_i^L$ for $SE(2)$ in polar coordinates as

$$\tilde{X}_1^L = -\frac{\partial}{\partial \phi} - \frac{\partial}{\partial \theta}$$
$$\tilde{X}_2^L = -\cos\theta\frac{\partial}{\partial r} + \frac{\sin\theta}{r}\frac{\partial}{\partial \theta}$$
$$\tilde{X}_3^L = -\sin\theta\frac{\partial}{\partial r} - \frac{\cos\theta}{r}\frac{\partial}{\partial \theta}.$$

D. Motion-Group Fourier Transform

The Fourier transform of a function of motion, $f(g)$, is an infinite-dimensional matrix defined as [2]

$$\mathcal{F}(f) = \hat{f}(p) = \int_G f(g) U(g^{-1}, p) d(g)$$

where $d(g)$ is a volume element at g, and $U(\cdot, p)$ is a unitary matrix function (called an irreducible unitary representation) for each value of the parameter p. The corresponding inverse Fourier transform (IFT) is

$$f(g) = \mathcal{F}^{-1}(f) = \int_{\hat{G}} \operatorname{trace}[\hat{f}(p) U(g,p)]\, d\nu(p) \tag{13}$$

where $\hat{G}$ is the space of all p values called the dual of the group G, and ν is an appropriately chosen integration measure in a generalized sense on $\hat{G}$.

For the case of $G = SE(2)$, the matrix elements of $U(g,p)$ are expressed explicitly as [30]

$$u_{mn}(g(\phi, r, \theta), p) = j^{n-m} e^{-j[n\phi + (m-n)\theta]} J_{n-m}(pr) \tag{14}$$

where $J_\nu(x)$ is the νth-order Bessel function, and $j = \sqrt{-1}$. The IFT (13) can be written explicitly for $SE(2)$ in terms of these matrix elements as

$$f(g) = \sum_{m,n\in\mathbf{Z}} \int_0^\infty \hat{f}_{mn} u_{nm}(g,p) p dp. \tag{15}$$

E. Operational Properties

In analogy with the classical Fourier transform, which converts derivatives of functions of position into algebraic operations in Fourier space, there are operational properties for the motion-group Fourier transform.

By the definition of the motion-group Fourier transform and differential operators, one observes that

$$\mathcal{F}\left(\tilde{X}_i^L f\right) = \int_G \frac{d}{dt}\left(f(\exp(-t\tilde{X}_i)\circ g)\right)\Big|_{t=0} U(g^{-1},p)d(g) \tag{16}$$

$$\mathcal{F}\left(\tilde{X}_i^R f\right) = \int_G \frac{d}{dt}(f(g\circ\exp(t\tilde{X}_i)))\Big|_{t=0} U(g^{-1},p)d(g). \tag{17}$$

By performing the change of variables $k = \exp(-t\tilde{X}_i)\circ g$ and using the homomorphism property of the representations $U(g,p)$, we can rewrite (16) as

$$\begin{aligned}\mathcal{F}\left(\tilde{X}_i^L f\right) &= \int_G f(k)\frac{d}{dt}U(k^{-1}\circ\exp(-t\tilde{X}_i),p)\Big|_{t=0} d(k)\\ &= \left(\int_G f(k)U(k^{-1},p)\,d(k)\right)\\ &\quad\times\left(\frac{d}{dt}U(\exp(-t\tilde{X}_i),p)\Big|_{t=0}\right)\\ &= -\hat{f}(p)\eta(\tilde{X}_i,p)\end{aligned}$$

where

$$\eta(\tilde{X}_i,p) = \left(\frac{d}{dt}U(\exp(t\tilde{X}_i),p)\right)\Big|_{t=0}. \tag{18}$$

Similarly, it can be shown that

$$\mathcal{F}\left(\tilde{X}_i^R f\right) = \eta(\tilde{X}_i,p)\hat{f}(p). \tag{19}$$

The explicit expression of $\eta(\tilde{X}_i,p)$ for $SE(2)$ can be derived as follows.

The matrix elements of $U(\exp(t\tilde{X}_1),p)$ can be obtained from (14) by setting $\phi = t, r = 0$, and $\theta = 0$

$$u_{mn}(\exp(t\tilde{X}_1),p) = e^{-jmt}\delta_{m,n}.$$

The fact that

$$J_{m-n}(0) = \begin{cases}1, & m-n=0\\ 0, & m-n\neq 0\end{cases}$$

is used in the above calculation. It then follows that

$$\eta_{mn}(\tilde{X}_1,p) = -jm\delta_{m,n}. \tag{20}$$

The matrix elements of $U(\exp(t\tilde{X}_2),p)$ can be obtained from (14) by setting $\phi = 0, r = t$, and $\theta = 0$

$$u_{mn}(\exp(t\tilde{X}_2),p) = j^{n-m}J_{n-m}(pt).$$

It is known that

$$\frac{d}{dx}J_m(x) = \frac{1}{2}[J_{m-1}(x) - J_{m+1}(x)].$$

Hence

$$\eta_{mn}(\tilde{X}_2,p) = \frac{jp}{2}(\delta_{m,n+1} + \delta_{m,n-1}). \tag{21}$$

The matrix elements of $U(\exp(t\tilde{X}_3),p)$ can be obtained from (14) by setting $\phi = 0, r = t, \theta = \pi/2$

$$u_{mn}(\exp(t\tilde{X}_3),p) = (-1)^{n-m}J_{n-m}(pt)$$

and so

$$\eta_{mn}(\tilde{X}_3,p) = \frac{p}{2}(\delta_{m,n+1} - \delta_{m,n-1}). \tag{22}$$

IV. Workspace Generation as a Diffusion Process

The concept and methods developed here can be used for spatial manipulators as well as planar ones. However, the planar case is used to introduce and explain our algorithms, because it is much easier to visualize concepts in the plane than in space.

A. Diffusion Equation for Workspace Generation

We obtain the diffusion equation for workspace generation by realizing that some characteristics of hyper-redundant manipulators are similar to those of polymer chains like DNA. During our study of conformational statistics in polymer science, we derived a diffusion equation defined on the motion group $SE(3)$ [3], [31]. This equation describes the evolution of a pdf for the position and orientation of the distal end of a stiff macromolecule relative to its proximal end. By incorporating parameters into this equation which indicate the kinematic properties of a manipulator, we can modify it to describe the evolution of the workspace density function. In the planar case, it is written explicitly as

$$\frac{\partial f}{\partial L} = \left(\alpha\tilde{X}_1^R + \beta\left(\tilde{X}_1^R\right)^2 + \tilde{X}_3^R + \epsilon\left(\tilde{X}_3^R\right)^2\right)f. \tag{23}$$

f stands for the workspace density function. L is the manipulator length. $\tilde{X}_1^R$ and $\tilde{X}_3^R$ are the differential operators defined on $SE(2)$ as given in Section III-C. Since the motion of a planar manipulator includes rotation around the Z axis and translation in the X-Y plane, only $\tilde{X}_1^R$ and $\tilde{X}_3^R$ are considered in (23), where the subscript 1 denotes the rotation around the Z axis, and the subscript 3 stands for the tangent direction along the manipulator's backbone. Parameters β and ϵ are usually called diffusion coefficients, and α is a drift coefficient. Here, we interpret them in the sense of the kinematic properties of manipulators. We define these kinematic properties as flexibility, extensibility, and the degree of asymmetry. β describes the flexibility of a manipulator in the sense of how much a segment of the manipulator can bend per unit length. A larger value of β means that the manipulator can bend a lot. ϵ describes the extensibility of a manipulator in the sense of how much a manipulator can extend along its backbone direction. A larger value of ϵ means that the manipulator can extend a lot. α describes the asymmetry in how the manipulator bends. When $\alpha = 0$, the manipulator can reach left and right with equal ease. When $\alpha < 0$, there is a preference for bending to the left, and when $\alpha > 0$, there is a

preference for bending to the right. Since α, β, and ϵ are qualitative descriptions of the kinematic properties of a manipulator, they are not directly measurable. In Section V, we will show how to choose the values of these parameters from the framework of probability theory for the best fit to a given manipulator.

B. Mathematical Validation of the Diffusion Model

If we cut any serial hyper-redundant manipulator of length $L_1 + L_2$ into two pieces of length L_1 and L_2, each with workspace densities $f(g; L_1)$ and $f(g; L_2)$, then the workspace density function $f(g; L_1 + L_2)$ is the motion-group convolution [4]

$$f(g; L_1 + L_2) = f(g; L_1) * f(g; L_2) \triangleq \int_G f(h; L_1) f(h^{-1} \circ g; L_2)\, dh \quad (24)$$

where $*$ denotes $SE(N)$ convolution. Here $L_i = n_i L_M$ where L_M is the length of one module, and n_i is the number of modules in segment i. The Fourier transform for $SE(N)$ converts this to a product as

$$\hat{f}(p; L_1 + L_2) = \hat{f}(p; L_2)\hat{f}(p; L_1)$$

where p is the "frequency" parameter. Keep in mind that this must hold no matter how we partition the manipulator. This indicates that each of these Fourier matrices must be of the form

$$\hat{f}(p; L) = \exp(LD(p))$$

where $D(p)$ is a matrix. Taking the partial derivative of both sides of the above equation with respect to L yields

$$\frac{\partial \hat{f}(p; L)}{\partial L} = D(p)\hat{f}(p; L).$$

Earlier in the paper, we showed that the differential operators $\tilde{X}_i^R$ transform under certain operational properties to matrices of the form $\eta(\tilde{X}_i, p)$. In addition, $SE(N)$ is an $M = N(N+1)/2$-dimensional Lie group, and any left-invariant differential operator $\tilde{D}$ can be constructed as

$$\tilde{D} = \sum_{i=1}^{M} a_i \tilde{X}_i^R + \sum_{i,j=1}^{M} b_{ij} \tilde{X}_i^R \tilde{X}_j^R + \sum_{i,j,k=1}^{M} c_{ijk} \tilde{X}_i^R \tilde{X}_j^R \tilde{X}_k^R + \cdots.$$

In essence, this is an analog of the Taylor series for a Lie group. If one takes enough terms, then the Fourier matrix of this Taylor series can be made to fit any $D(p)$ by appropriate choice of the coefficients a_i, b_{ij}, etc. We truncate at relatively few parameters (based on our experience in an example in polymer theory, where only a relatively few parameters have physical meaning). However, the true test is the goodness with which we can fit the example later in this paper. If these few parameters were not sufficient, we could take more. The parameters defining the details of the diffusion can be changed to meet different manipulator geometries as needed.

C. Solution to the Diffusion Equation

The procedure to solve the diffusion equation is explained in Fig. 3. We first apply the motion-group Fourier transform to the diffusion equation (23). Using the operational properties (19),

$$\frac{\partial f}{\partial L} = \left(\alpha \tilde{X}_1^R + \beta(\tilde{X}_1^R)^2 + \tilde{X}_3^R + \varepsilon(\tilde{X}_3^R)^2\right) f$$

↓ Motion-group Fourier Transform

$$\frac{d\hat{f}}{dL} = B(p)\,\hat{f}$$

↓ Matrix Exponential

$$\hat{f}(p; L) = \exp(L\,B(p))$$

↓ Motion-group Inverse Fourier Transform

$$f(g; L) = F^{-1}(\hat{f})$$

Fig. 3. Solving method to the diffusion equation (23).

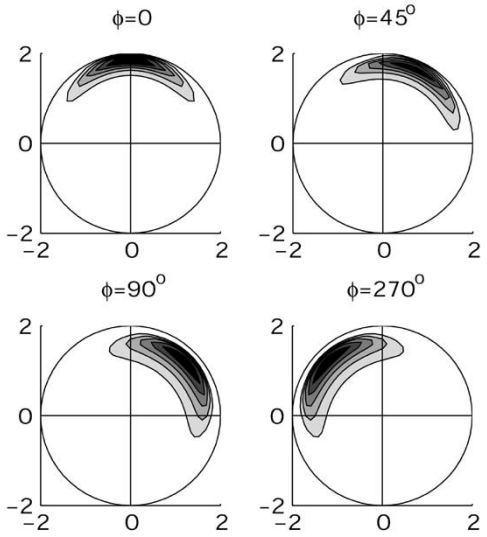

Fig. 4. Workspace density at different orientations.

the diffusion equation (23) is converted to a system of linear ordinary differential equations with constant coefficients

$$\frac{d\hat{f}}{dL} = B(p)\hat{f} \quad (25)$$

where

$$B = \alpha\eta(\tilde{X}_1, p) + \beta[\eta(\tilde{X}_1, p)]^2 + \eta(\tilde{X}_3, p) + \epsilon[\eta(\tilde{X}_3, p)]^2.$$

Explicit expressions for the matrix elements of $\eta(\tilde{X}_1, p)$ and $\eta(\tilde{X}_3, p)$ are derived in (20) and (22), respectively.

In principle, $f(g; 0) = \delta(g)$ (the Dirac delta function is for $SE(2)$), and $\hat{f}(p; 0)$ is the identity. The solution to (25) can be obtained by the matrix exponential

$$\hat{f}(p; L) = \exp(LB(p)). \quad (26)$$

Then we can get the solution $f(g; L)$ to the diffusion equation (23) by applying the motion-group IFT (15) to (26). Example solutions to the diffusion equation (23) at different orientation ϕ with $L = 2, \alpha = 0, \beta = 1$, and $\epsilon = 0$ are illustrated in Fig. 4.

The effects of the parameters in (23) can be observed obviously in the positional workspace densities. By integrating $f(g; L)$ over all values of rotation angle ϕ, we can obtain the positional workspace density as shown in Fig. 5. Fig. 5(a) and (b) show the effect of the length parameter L with $L = 1, 2$. Fig. 5(b) and (c) shows the effect of the extensibility parameter ϵ with $\epsilon = 0, 0.04$. The area of the workspace density is extended/fattened because of the larger value of ϵ. Fig. 5(b) and (d)

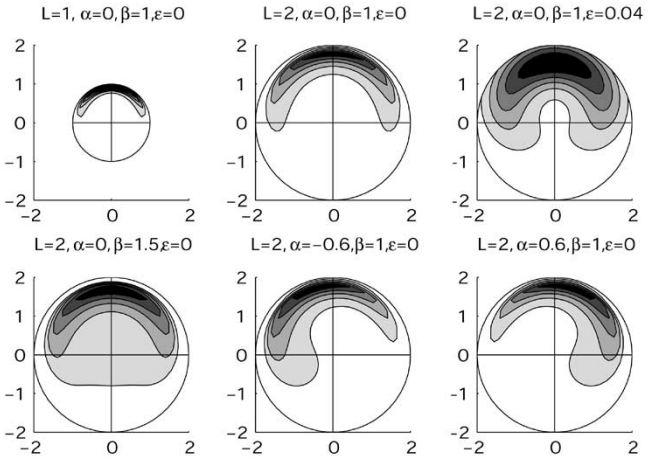

Fig. 5. Effects of parameters L, α, β, and ϵ.

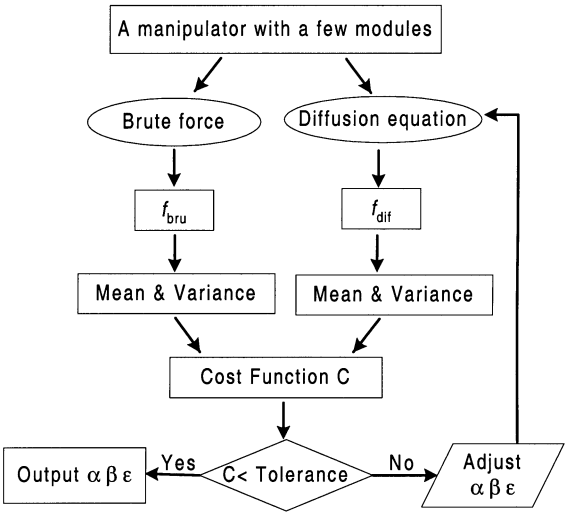

Fig. 6. Flowchart of the matching method.

shows the effect of the flexibility parameter β with $\beta = 1, 1.5$. For a larger value of β, the workspace density is supported with a larger area. Fig. 5(b), (e), and (f) shows the effect of the asymmetry parameter α with $\alpha = 0, -0.6, 0.6$. It is easy to notice how α affects the workspace of a manipulator to bend to the left and right.

In the above numerical implementations, the infinite-dimensional matrix function $U(g, p)$ is truncated. The result is a bandlimited approximation. We chose the upper bound of the frequency parameter p to be 100. The matrix $U(g, p)$ is truncated at $-l_B \leq m, n \leq l_B$, where $l_B = 7$. Since the numerical results of the Fourier transform of this diffusion equation are approximated by a bandlimited version, the outer elements (values of $\hat{f}(p; L) = \exp(B(p)L)$ with $|m|, |n| \to l_B$) can have larger errors. We therefore impose a second cutoff frequency of $l_B = 4$ after the exponentiation, when substituting into the Fourier inverse formula to obtain the workspace density function $f(g; L)$. All these numerical results are calculated using Matlab on a Pentium III 64MB RAM PC. It takes no more than 40 s to solve the diffusion equation (23) and get the workspace density.

V. Choice of Parameters in the Algorithm

Different manipulators have different kinematic properties. It is impossible to find a closed-form relationship between the parameters α, β, and ϵ and the kinematic properties of manipulators that is suitable for all manipulators. Also, these parameters are a qualitative description of the kinematic properties and not directly measurable. We developed a general approach based on probability theory to match the parameters α, β, and ϵ to a given manipulator.

The proposed matching method uses the mean and variance of workspace density functions. The general idea is to adjust α, β, and ϵ to make the mean and variance of the workspace density functions obtained from (23) and brute-force enumeration as similar as they can be. The values of α, β, and ϵ characterizing the manipulators are the ones that make the workspace density functions have nearly the same mean and variance. The procedure for this matching method is depicted by the flowchart shown in Fig. 6.

For a given manipulator, we consider only a few modules, so that its workspace can be easily obtained by brute-force enumeration. We first calculate the workspace density function f_{bru} using brute-force enumeration, and f_{dif} using the diffusion equation (23). The initial values to the parameters α, β, and ϵ in (23) are assigned manually by observing features in the shape of f_{bru}. Then we calculate the corresponding mean in terms of x, y, ϕ, and variances in terms of $x^2, y^2, \phi^2, xy, x\phi, y\phi$. The parameters α, β, and ϵ are adjusted to minimize the cost function

$$\begin{aligned} C(\alpha, \beta, \epsilon) = {} & (\tilde{E}_x - E_x)^2 + (\tilde{E}_y - E_y)^2 + (\tilde{E}_\phi - E_\phi)^2 \\ & + \left(\tilde{\sigma}_x^2 - \sigma_x^2\right)^2 + \left(\tilde{\sigma}_y^2 - \sigma_y^2\right)^2 + \left(\tilde{\sigma}_\phi^2 - \sigma_\phi^2\right)^2 \\ & + \left(\tilde{\sigma}_{xy}^2 - \sigma_{xy}^2\right)^2 + \left(\tilde{\sigma}_{x\phi}^2 - \sigma_{x\phi}^2\right)^2 \\ & + \left(\tilde{\sigma}_{y\phi}^2 - \sigma_{y\phi}^2\right)^2 \end{aligned} \tag{27}$$

where the $\tilde{E}$'s and $\tilde{\sigma}$'s are the means and variances of f_{dif}, and the E's and σ's are the means and variances of f_{bru}.

There are several ways to adjust α, β, and ϵ. Since it is easy to find the coarse range of these parameters, one simple way is to enumerate all the possible values with a small incremental step of these parameters for a given range, and find the set of parameters that results in the minimal value of the cost function. Our numerical simulations show that the value of cost function is not sensitive to small changes in these parameters, so the step of the increment will not affect the final result significantly.

VI. Numerical Simulations for Workspace Generation

The manipulator used for the numerical simulation is the same as the one that has the workspace shown in Fig. 2. We use this manipulator with four modules to match the parameters. The maximal distance between a reachable position and the proximal end (origin) for this manipulator with four modules is six. Hence, the length parameter L in (23) is set to be six. Since the workspace is symmetric in bending to the left and right, the parameter α is set to be zero. The range of parameter β is taken from 0.02 to 0.74 with the increment step of 0.02. The

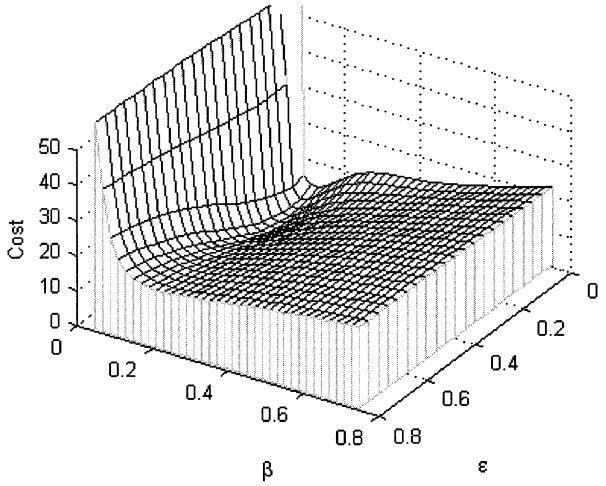

Fig. 7. Values of the cost function.

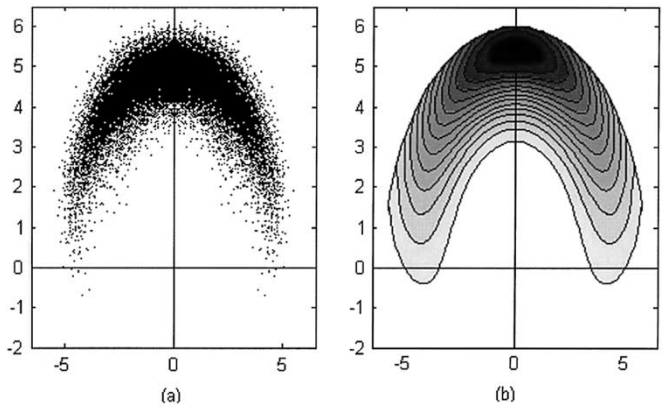

Fig. 8. (a) Workspace and (b) its density of a four-module manipulator.

range of parameter ϵ is taken from 0 to 0.8 with the increment step of 0.04. Fig. 7 shows the corresponding values of the cost function (27). We found that the minimal value of the cost function is 0.5748 when $\beta = 0.12$ and $\epsilon = 0.08$. Hence, the parameters characterizing this manipulator are taken to be $\alpha = 0, \beta = 0.12$, and $\epsilon = 0.08$. The discrete workspace and the corresponding density of this manipulator with four modules generated from the diffusion equation (23) with $L = 6, \alpha = 0, \beta = 0.12$, and $\epsilon = 0.08$ are shown in Fig. 8(a) and (b), respectively.

Equation (24) indicates the linear relationship between the length and number of modules of a manipulator in the workspace density function. In our simulation, we use a four-module manipulator to match the parameters α, β, and ϵ in the diffusion equation (23) with the length parameter $L = 6$. Because of the linear relationship shown in (24), the workspace density produced by adding or reducing one module of a manipulator is the same as that by increasing or decreasing the length parameter L by 1.5 (i.e., the length corresponding to one module $=$ length of $6/4$ modules). To verify this result, we test the manipulator with five modules using the matched parameters $\alpha = 0, \beta = 0.12$, and $\epsilon = 0.08$. Fig. 9(a) and (b) show the discrete workspace and the corresponding density of this manipulator with five modules generated from the diffusion equation (23) with $L = 7.5$, respectively.

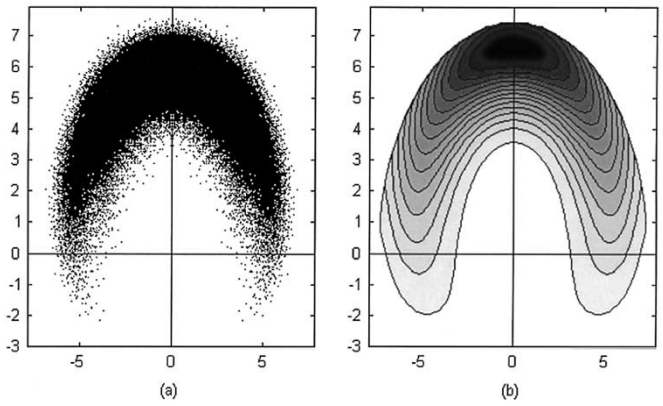

Fig. 9. (a) Workspace and (b) its density of a five-module manipulator.

Of course, the benefit of using the diffusion approach is that the cost is the same for any length L, whereas brute force goes as $K^{\lambda L}$, where λ is number of modules per unit length, and K is number of sample points per module.

When the modules of a manipulator are not all identical, a different diffusion equation is solved for each segment of the manipulator composed of the same kind of modules, then the convolution of the different segments is performed as matrix multiplication in Fourier space, and the Fourier inversion formula is used to recover the workspace density.

VII. Inverse Kinematics Using Workspace Density

To further confirm our diffusion-based workspace generation algorithm, we apply the workspace density generated by the diffusion equation to solve the inverse kinematics problem. The way we apply the workspace density to inverse kinematics is similar to the Ebert–Uphoff method [2]. The criterion for this method is to configure the manipulator to achieve the maximum workspace density around the target spot.

A. Scheme for Using Workspace Density

Consider a manipulator with P independent modules. Let g_k denote the homogenous transformation that relates the distal end of the kth module to its own base, where $k \in \{1, \ldots, P\}$. The transformation of the distal end of the kth module relative to the base of the whole manipulator is then

$$g^{(k)} = g_1 \circ g_2 \circ \cdots \circ g_k$$

and the transformation that relates the distal end of the manipulator to the distal end of the kth module is

$$\left(g^{(k)}\right)^{-1} \circ g^{(P)} = g_{k+1} \circ g_{k+2} \circ \cdots \circ g_P.$$

Let $f_{i-1,i}(g)$ denote the workspace density function for the ith module. With this notation, $f_{0,P}(g) = f(g; L)$ will be the workspace density function for the whole manipulator. The workspace density function for two concatenated modules is the convolution [4]

$$f_{i-1,i+1}(g) = (f_{i-1,i} * f_{i,i+1})(g)$$

and the workspace density function for the whole manipulator is the multiple convolution

$$f_{0,P}(g) = (f_{0,1} * f_{1,2} * \cdots * f_{P-1,P})(g).$$

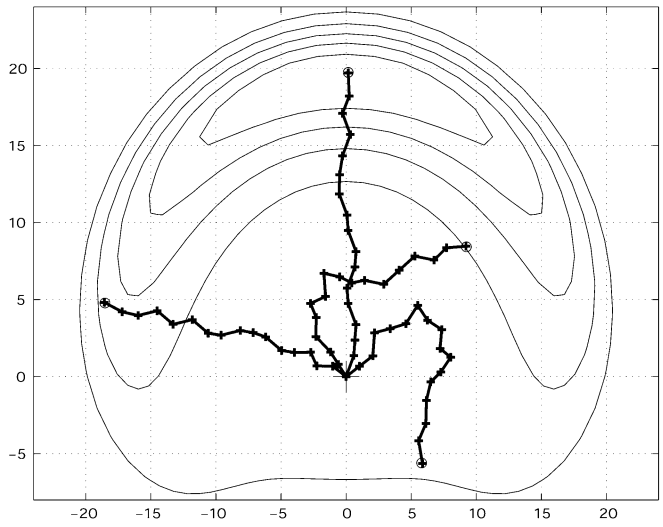

Fig. 10. Inverse kinematics solutions for the 16-module manipulator.

Using the above fact, we divide the manipulator into virtual segments and view the lower segment as a transport device for the workspace density of the upper segment. We then fix the configuration of the lower segment by choosing the one that results in the highest density of the upper segment at the target spot.

B. Inverse Kinematics Method

There are several ways to realize the above scheme. One way is to start from the base module and proceed up one module sequentially until the last module, as the Ebert–Uphoff method [2] does. Let g_{des} denote the target spot. Starting from the base module, we transport the base module to all possible states, and find which state of this module maximizes the density $f_{P-1}((g^{(1)})^{-1} \circ g_{\mathrm{des}})$. Suppose we find $g_{1,\mathrm{opt}}$ is such a state. We fix the base module to $g_{1,\mathrm{opt}}$. Then we proceed up the manipulator one module. Among all possible states of $g^{(2)} = g_{1,\mathrm{opt}} \circ g_2$, we search for the state of g_2 that can achieve the highest density $f_{P-2}((g^{(2)})^{-1} \circ g_{\mathrm{des}})$. If $g_{2,\mathrm{opt}}$ is such a state, we configure the second module to $g_{2,\mathrm{opt}}$. This procedure is performed by sequentially maximizing $f_{P-k}((g^{(k)})^{-1} \circ g_{\mathrm{des}})$ for all $k \in \{1, 2, \ldots, P-2\}$. When $k = P$, the state of g_P that minimizes the cost function

$$C = D(g_{\mathrm{des}}, g_1 \circ g_2 \circ \cdots \circ g_P)$$

is chosen, where $D(\cdot, \cdot)$ is a distance metric [17], [33].

The novelty of this inverse kinematics approach lies in the fact that the functions $f_k(g) = f(g; (k/N)L)$ can all be computed without explicitly performing convolution on $SE(2)$. Rather, we only solve the diffusion equation (23) for different length values of kL/N.

C. Numerical Simulations for Inverse Kinematics

The sample manipulator with 16 modules is used to simulate the inverse kinematics approach. Targets located in various areas of the workspace are tested, and four of them are shown in Fig. 10. The contour plot on the background of the figure indicates the workspace of this manipulator with 16 modules. The circles stand for the targets. The cross denotes the position of the fixed end of the manipulator. The segmented lines display the corresponding configurations of the manipulator, where each segment stands for a module. From all these tests, we see that the inverse kinematics algorithm using the workspace density generated by the diffusion equation (23) provides an accurate solution to reach the target.

The above simulations are implemented using Matlab on a Pentium III 64MB RAM PC. It takes less than 1 s to compute the inverse kinematics solutions. Note that the computation time for solving the diffusion equation is not counted, since we store the workspace density of this manipulator with different numbers of modules in advance.

VIII. Conclusion

Workspace generation is an important issue for applications of hyper-redundant manipulators to be realized. In this paper, we have shown that the workspace of a hyper-redundant manipulator can be generated by solving a partial differential equation defined on the motion group. The computational complexity of this workspace generation algorithm is independent of the number of modules/DOFs of a manipulator. In this sense, this approach is very suitable for hyper-redundant manipulators. To our knowledge, this is the only method in which computational complexity is not affected by the number of modules/DOFs. We have also shown that the workspace generated by the diffusion equation can be applied to perform inverse kinematics efficiently.

References

[1] D. G. Alciatore and C. C. D. Ng, "Determining manipulator workspace boundaries using the Monte Carlo method and least square segmentation," *ASME Robot.: Kinematics, Dynam., Controls*, vol. DE-72, pp. 141–146, 1994.

[2] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton, FL: CRC Press, 2000.

[3] G. S. Chirikjian and Y. F. Wang, "Conformational statistics of stiff macromolecules as solutions to PDE's on the rotation and motion groups," *Phys. Rev. E*, vol. 62, no. 1, pp. 880–892, July 2000.

[4] G. S. Chirikjian and I. Ebert–Uphoff, "Numerical convolution on the Euclidean group with applications to workspace generation," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 123–136, Feb. 1998.

[5] G. S. Chirikjian and J. W. Burdick, "A hyper-redundant manipulator," *IEEE Robot. Automat. Mag.*, vol. 1, pp. 22–29, Dec. 1994.

[6] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *J. Aerosp. Eng.*, vol. 12, no. 2, pp. 65–73, 1999.

[7] R. Cieslak and A. Morecki, "Elephant trunk-type elastic manipulator: A tool for bulk and liquid materials transportation," *Robotica*, vol. 17, pp. 11–16, 1999.

[8] J. J. Craig, *Introduction to Robotics, Mechanics and Control*. Reading, MA: Addison-Wesley, 1989.

[9] I. Ebert–Uphoff, "On the development of discretely-actuated hybrid-serial-parallel manipulators," Ph.D. dissertation, Dept. Mech. Eng., Johns Hopkins Univ., Baltimore, MD, 1997.

[10] I. M. Gelfand, R. A. Minlos, and Z. Y. Shapiro, *Representations of the Rotation and Lorentz Groups and Their Applications*. New York: Pergamon, 1963.

[11] I. A. Gravagne and I. D. Walker, "On the kinematics of remotely actuated continuum robots," in *Proc. Int. Conf. Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 2544–2550.

[12] M. W. Hannan and I. D. Walker, "The elephant trunk manipulator, design and implementation," in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, Como, Italy, July 2001, pp. 14–19.

[13] S. Hirose, *Biologically Inspired Robots*. Oxford, U.K.: Oxford Univ. Press, 1993.

[14] S. Hirose and S. Ma, "Coupled tendon-driven multijoint manipulator," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 1268–1275.

[15] M. Ivanescu and V. Stoian, "A variable structure controller for a tentacle manipulator," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 3155–3160.
[16] K. Koganezawa and T. Kinoshita, "Hyper-redundant manipulator using compound three-bar linkages," in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, Como, Italy, July 2001, pp. 8–13.
[17] K. Kazerounian and J. Rategar, "Object norms: A class of coordinate and metric independent norms for displacement," *Flexible Mechan., Dynam., Anal.*, vol. DE-47, pp. 271–275, 1997.
[18] S. J. Kwon, Y. Youm, and W. K. Chung, "General algorithm for automatic generation of the workspace for n-link planar redundant manipulators," *ASME Trans.*, vol. 116, pp. 967–969, 1994.
[19] A. B. Kyatkin and G. S. Chirikjian, "Synthesis of binary manipulators using the Fourier transform on the Euclidean group," *ASME J. Mech. Des.*, vol. 121, pp. 9–14, Mar. 1999.
[20] S. Ma, S. Hirose, and H. Yoshinada, "Development of a hyper-redundant manipulator for maintenance of nuclear reactors," *Int. J. Adv. Robot.*, vol. 9, no. 3, pp. 281–300, 1995.
[21] S. Ma and M. Konno, "An obstacle avoidance scheme for hyper-redundant manipulators: Global motion planning in posture space," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 161–166.
[22] S. Ma and M. Watanabe, "Minimum-time control of coupled tendon-driven manipulators," *Int. J. Adv. Robot.*, vol. 15, no. 4, pp. 409–427, 2001.
[23] W. Miller, *Lie Theory and Special Functions*. New York: Academic, 1968.
[24] H. Mochiyama, E. Shimemura, and H. Kobayashi, "Shape control of manipulators with hyper degrees of freedom," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 584–600, 1999.
[25] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
[26] E. Paljug, T. Ohm, and S. Hayati, "The JPL serpentine robot: A 12-DOF system for inspection," in *Proc. Int. Conf. Robotics and Automation*, 1995, pp. 3143–3148.
[27] J. Rastegar and P. Deravi, "Methods to determine workspace, its subspaces with different numbers of configurations and all the possible configurations of a manipulator," *Mech. Mach. Theory*, vol. 22, no. 4, pp. 343–350, 1987.
[28] R. Ricard and C. Gosselin, "On the determination of the workspace of complex planar robotic manipulators," in *Proc. ASME Mechanisms Conf.*, vol. DE-72, Sept. 1994, pp. 133–140.
[29] D. Sen and T. S. Mruthyunjaya, "A discrete state perspective of manipulator workspaces," *Mech. Mach. Theory*, vol. 29, no. 4, pp. 591–605, 1994.
[30] N. J. Vilenkin and A. U. Klimyk, *Representations of Lie Groups and Special Functions*. Amsterdam, The Netherlands: Kluwer, 1991, vol. 1–3.
[31] Y. F. Wang, "Applications of diffusion processes in robotics, optical communications and polymer science," Ph.D. dissertation, Dept. Mech. Eng., Johns Hopkins Univ., Baltimore, MD, 2001.
[32] K. E. Zanganeh, R. S. K. Lee, and P. C. Hughes, "A discrete model for the configuration control of hyper-redundant manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp. 167–172.
[33] M. Zefran, V. Kumar, and C. Croke, "Metrics and connections for rigid-body kinematics," *Int. J. Robot. Res.*, vol. 18, no. 2, pp. 243–258, 1999.

Yunfeng Wang (M'04) received the B.S. and M.S. degrees in mechanical engineering from Tianjin University, Tianjin, China, in 1992 and 1995, respectively, and the M.S.E. degree in electrical and computer engineering in 2001, and the Ph.D. degree in mechanical engineering in 2002, both from The Johns Hopkins University, Baltimore, MD.

She is currently an Assistant Professor in the Department of Engineering, The College of New Jersey, Ewing, NJ. Her research interests include hyper-redundant manipulators, mechatronics, and intelligent manufacturing.

Gregory S. Chirikjian (M'93) was born August 16, 1966, in New Brunswick, NJ. He received the B.S.E. degree in engineering mechanics, the M.S.E. degree in mechanical engineering, and the B.A. degree in mathematics, all from The Johns Hopkins University, Baltimore, MD, in 1988. He then received the Ph.D. degree from the California Institute of Technology, Pasadena, CA, in 1992.

Since 1992, he has been on the faculty of the Department of Mechanical Engineering, The Johns Hopkins University, where he is now a Professor. His research interests include the kinematic analysis, motion planning, design, and implementation of hyper-redundant, metamorphic, and binary manipulators. In recent years, he has expanded the scope of his research to include applications of group theory in a variety of engineering disciplines and the mechanics of biological macromolecules.

Dr. Chirikjian is a 1993 National Science Foundation Young Investigator, a 1994 Presidential Faculty Fellow, and a 1996 recipient of the ASME Pi Tau Sigma Gold Medal.