Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

A Diffusion-based Algorithm for Workspace Generation of Highly Articulated Manipulators

Yunfeng Wang, Gregory S. Chirikjian
Department of Mechanical Engineering
Johns Hopkins University
Baltimore, MD 21218, USA
Email: {yfwang, gregc}@jhu.edu

Abstract

Motivated by a physical phenomenon, the diffusion process, this paper develops a diffusion-based algorithm for workspace generation of highly articulated manipulators. This algorithm makes the workspace generation problem as simple as solving a diffusion-type equation which has an explicit solution. This equation is a partial differential equation defined on the motion group and describes the evolution of the workspace density function depending on manipulator length and kinematic properties. Numerical simulations using this algorithm are also presented.

1 Introduction

Highly articulated manipulators, which are also called snake-like, serpentine or hyper-redundant manipulators, possess agility 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. Efficient algorithms for workspace generation are important for applications of such manipulators in the real world.

The workspace generation of a highly articulated manipulator is much more complicated than that of a manipulator with a few links. Various methods have been proposed. Geometric methods were used by [6] to generate the workspace for a manipulator with an arbitrary number of revolute joints. However, such algorithms are not applicable for manipulators of general structure. A curve approximation approach is presented to determine the workspace of complex planar manipulators in [9]. The authors of [8] divided the manipulator into parts, subworkspaces of which are then calculated using the Jacobian. The Monte-Carlo method was used in [1], where a large number of random actuator values are generated and the corresponding reachable positions are calculated. In terms of workspace density, the authors of [5] 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 parts, and approximating the workspace of each part as a density function.

Different from all the above mentioned methods, this paper develops an approach to generate the workspace of a highly articulated manipulator by solving a partial differential equation defined on the motion group, $SE(N)$. This equation can be easily solved using the techniques of Fourier analysis of motion as developed in [2].

The format of the remainder of this paper is as follows. In Section 2, the required mathematical techniques are reviewed. The diffusion-based algorithm is presented in Section 3. Section 4 explains a method to choose kinematic parameters in the proposed diffusion-type equation for a given manipulator. Section 5 provides the simulation results.

2 Fourier Analysis of Motion

2.1 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 $g = (\mathbf{a}, A)$ where $A \in SO(N)$ and $\mathbf{a} \in \mathbb{R}^N$. For any $g = (\mathbf{a}, A)$ and $h = (\mathbf{r}, R) \in SE(N)$, the group law is written as $g \circ h = (\mathbf{a} + A\mathbf{r}, AR)$, and $g^{-1} = (-A^T\mathbf{a}, A^T)$. It is often convenient to express an element of $SE(N)$ as an homogeneous transformation matrix of the form:

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

[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.

0-7803-7272-7/02/$17.00 © 2002 IEEE

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

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

$SE(2)$ is a 3-dimensional manifold much like $I\!R^3$. We can integrate over $SE(2)$ using the volume element $d(g(r,\theta,\phi)) = \frac{1}{4\pi^2} r dr d\theta d\phi$ [2].

2.2 Differential Operators Defined on the Motion Group

The way to take partial derivatives of a function of motion is to evaluate

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

where $\tilde{X}_i$ is the basis for the Lie algebra. For the case of $SE(2)$,

$$\tilde{X}_1 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix};$$

$$\tilde{X}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix};$$

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

Explicitly, we can derive the differential operators $\tilde{X}_i^R$ for $SE(2)$ in polar coordinates to be [10]

$$\begin{aligned} \tilde{X}_1^R &= \frac{\partial}{\partial\phi} \\ \tilde{X}_2^R &= \cos(\phi-\theta)\frac{\partial}{\partial r} + \frac{\sin(\phi-\theta)}{r}\frac{\partial}{\partial\theta} \\ \tilde{X}_3^R &= -\sin(\phi-\theta)\frac{\partial}{\partial r} + \frac{\cos(\phi-\theta)}{r}\frac{\partial}{\partial\theta} \end{aligned} \quad (1)$$

2.3 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 $U(g,p)$ is an infinite dimensional matrix function of g and a frequency parameter p. The corresponding inverse Fourier transform is

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

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 $SE(2)$, the matrix elements of $U(g,p)$ are expressed explicitly as [2]:

$$u_{mn}(g(r,\theta,\phi),p) = j^{n-m} e^{-j[n\phi+(m-n)\theta]} J_{n-m}(pr) \quad (2)$$

where $J_\nu(x)$ is the ν^{th} order Bessel function. The inverse Fourier transform can be written in terms of elements as

$$f(g) = \sum_{m,n\in Z} \int_0^\infty \hat{f}_{mn} u_{nm}(g,p) p dp. \quad (3)$$

2.4 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 $\tilde{X}_i^R$, we can derive the Fourier transform of the derivatives of a function of motion as:

$$\mathcal{F}[\tilde{X}_i^R f] = \eta(\tilde{X}_i, p)\hat{f}(p)$$

where

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

Now we will derive the explicit expression of $\eta(\tilde{X}_i, p)$ for $SE(2)$ since they will be used to solve the partial differential equation stating in the next section.

The matrix elements of $U(\exp(t\tilde{X}_1), p)$ can be obtained from (2) by setting $\phi = t$, $r = 0$, and $\theta = 0$:

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

The fact

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

is used in the above calculation. Explicitly,

$$\eta_{mn}(\tilde{X}_1, p) = -jm\delta_{m,n}. \quad (4)$$

The matrix elements of $U(\exp(t\tilde{X}_2), p)$ can be obtained from (2) 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,

$$\eta_{mn}(\tilde{X}_2, p) = \frac{jp}{2}(\delta_{m,n+1} + \delta_{m,n-1}). \quad (5)$$

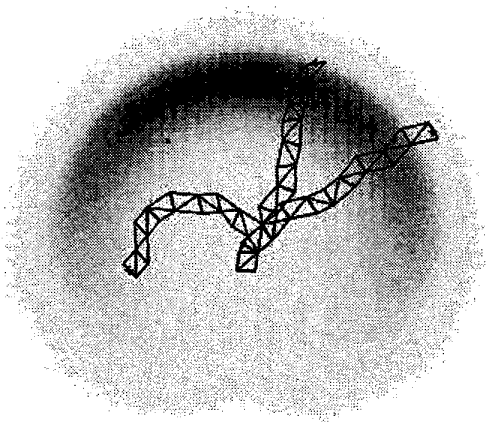

Figure 1: The workspace density of a binary manipulator

The matrix elements of $U(\exp(t\tilde{X}_3), p)$ can be obtained from (2) 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

$$\eta_{mn}(\tilde{X}_3, p) = \frac{p}{2}(\delta_{m,n+1} - \delta_{m,n-1}). \qquad (6)$$

3 Workspace Generation as a Diffusion Process

3.1 Workspace Density

The diffusion-based algorithm takes advantage of the concept of workspace density. Workspace density describes the density of reachable points/frames in any portion of the workspace [5] 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. The workspace density function is a probability density function that describes the distribution of points/frames over the workspace. Higher density of a point/frame means the manipulator can reach that point/frame more accurately. Figure 1 displays the workspace density of a binary manipulator with ten modules [4]. Three of the manipulator's 2^{30} configurations are indicated. Darker areas mean higher density, and manipulators can reach that part more accurately.

3.2 Inspiration of the Algorithm

Consider a discretely-actuated serial manipulator. Each module can reach 16 different states as shown in Figure 2. The workspace of this manipulator with 2 modules, 3 modules and 4 modules are also shown in Figure 2 respectively. They are generated by brute force enumeration. It is easy to notice that the size of the workspace spreads out with the increment of modules. 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 highly articulated 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.

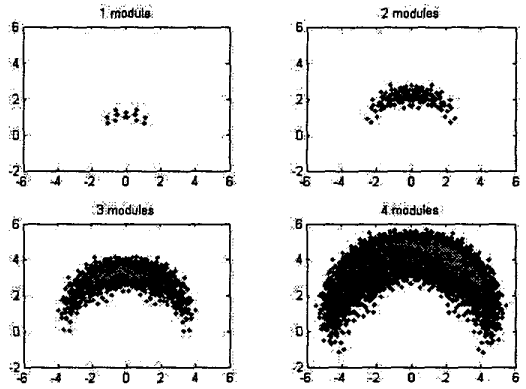

Figure 2: The workspace of a manipulator with different modules

3.3 Implementation of the Algorithm

With this analogy, the next step is to determine what kind of diffusion equation is suitable to model this process. We can obtain such an equation by realizing that some characteristics of highly articulated manipulators are similar to those of polymer chains like DNA.

During our study of conformational statistics in polymer science, we derived a diffusion-type equation defined on the motion group $SE(3)$ [3, 10]. This equation describes the evolution of a probability density function 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

$$\frac{\partial f}{\partial L} = \left(\alpha\tilde{X}_1^R + \beta(\tilde{X}_1^R)^2 + \tilde{X}_3^R + \epsilon(\tilde{X}_3^R)^2\right) f. \qquad (7)$$

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 $SE(2)$ where the subscript 1 denotes the rotation around Z-axis and the subscript 3 stands for the tangent direction along the manipulator's backbone. Parameters α, β and ϵ describe 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 description of the kinematic properties of a manipulator, they are not directly measurable. In Section 4, we will show how to choose the values of these parameters from the framework of probability theory.

This simple three-parameter model qualitatively captures the behavior that has been observed in numerical simulations of workspace densities of discretely-actuated variable-geometry truss manipulators [7].

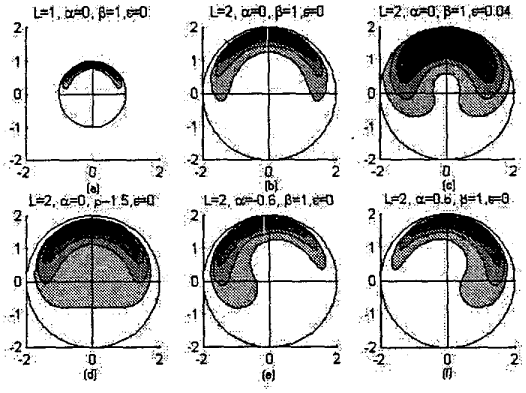

Figure 3: The effects of parameters L, α, β and ϵ

Explicit Solution to (7)

Applying the motion-group Fourier transform to (7) and utilizing its operational properties, we can convert (7) to an infinite system of linear ordinary differential equations with constant coefficients:

$$\frac{d\hat{f}}{dL} = B\hat{f} \tag{8}$$

where the matrix

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

The explicit expressions of the matrix elements of $\eta(\tilde{X}_1, p)$ and $\eta(\tilde{X}_3, p)$ are derived as (4) and (6) respectively.

In principle, $f(g; 0) = \delta(g)$, and $\hat{f}(p; 0)$ is the identity. The solution to (8) can be obtained by matrix exponential

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

Then we substitute the solution $\hat{f}(p; L)$ into the motion-group Fourier inversion formula (3) to recover $f(g; L)$.

The effects of the parameters in (7) can be shown in the positional workspace densities. The positional workspace density is obtained by integrating $f(g; L)$ over all values of rotation angle. Figures 3 (a) and (b) show the effect of the length parameter L with $L = 1, 2$. Figures 3 (b) and (c) show the effect of the extensibility parameter ϵ with $\epsilon = 0, 0.04$. The area of the workspace density is extended/fatted because of the larger value of ϵ. Figures 3 (b) and (d) show the effect of the flexibility parameter β with $\beta = 1, 1.5$. For larger value of β, the workspace density has larger area. Figures 3 (b), (e) and (f) show 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 band-limited 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 band-limited version, the outer elements (values of $\hat{f} = \exp(B)$ with $|m|, |n| \rightarrow L_B$) 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 $f(g; L)$.

4 The 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 any manipulator. 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 mean and variance of workspace density functions are used in this matching method. The general idea is to adjust α, β and ϵ to make the mean and variance of the workspace density functions obtained from (7) and brute force enumeration as similar as they can be. The 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 Figure 4.

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

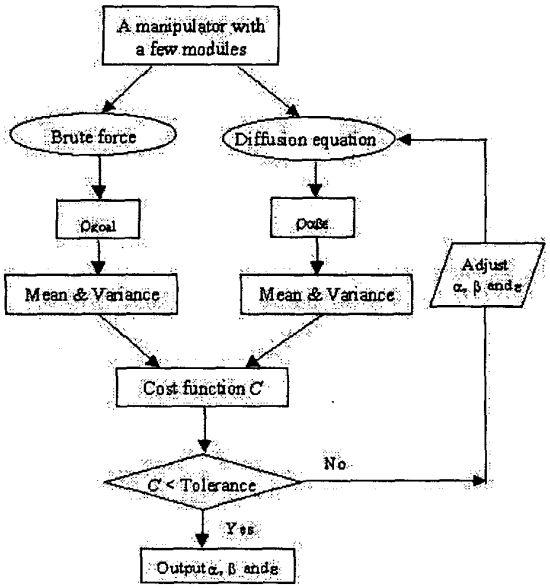

Figure 4: The flowchart of the matching method

function ρ_{goal} using brute force enumeration and $\rho_{\alpha,\beta,\epsilon}$ using the diffusion-type equation (7). The initial values to the parameters α, β, and ϵ in (7) are assigned through the features on the shape of ρ_{goal}. Then we calculate the corresponding mean in term of x, y, ϕ and variance in term of $x, y, \phi, xy, x\phi, y\phi$. The parameters α, β and ϵ are adjusted to minimize the cost function

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

where $\tilde{E}$'s and $\tilde{\sigma}$'s are the means and variances of $\rho_{\alpha,\beta,\epsilon}$ and E's and σ's are the means and variances of ρ_{goal}.

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 increment 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 the small change of these parameters, so the step of the increment will not affect the final result significantly.

5 Numerical Simulations

The manipulator used for the numerical simulation is the same as that shown in Figure 2. We use this manipulator with 4 modules to match the parameters. From Figure 2, we know that the maximal radius of the workspace of

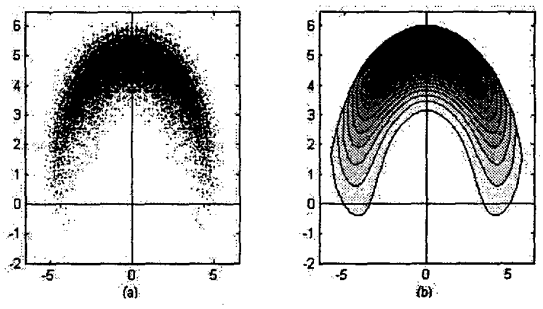

Figure 5: The workspace (a) and workspace density (b) of a 4-module manipulator

this manipulator with 4 modules is about 6. Hence, the parameter L in (7) is set to be 6. Since the workspace is symmetric in bending to the left and right, the parameter α is set to be 0. The range of parameter β is taken from 0.02 to 0.74 with the increment step of 0.02. The range of parameter ϵ is taken from 0 to 0.8 with the increment step of 0.04. 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 $\alpha = 0$, $\beta = 0.12$ and $\epsilon = 0.08$. The workspace and workspace density of this manipulator with 4 modules generated from the brute force and the diffusion-type equation (7) with $L = 6$, $\alpha = 0$, $\beta = 0.12$ and $\epsilon = 0.08$ are shown in Figures 5 (a) and (b) respectively.

It is shown in [4] that the workspace density function $\rho(g; L)$ for two concatenated manipulator segments with length L_1 and L_2 is the motion-group convolution

$$\rho(g; L_1 + L_2) = \rho(g; L_1) * \rho(g; L_2). \qquad (9)$$

This 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 4-module manipulator to match the parameters α, β and ϵ in the diffusion-type equation (7) with the length parameter $L = 6$. Because of the linear relationship shown in (9), 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 for the diffusion-type equation (7). To verify this result, we test the manipulator with 5 modules using the matched parameters $\alpha = 0$, $\beta = 0.12$ and $\epsilon = 0.08$. Figures 6 (a) and (b) show the workspace and workspace density of this manipulator with 5 modules generated from brute force method and the diffusion-type equation (7) with $L = 7.5$ respectively.

All the above numerical simulations can be implemented in a couple of minutes using a pentium III PC with 128 KB memory. We also further applied the workspace density generated in our algorithm to solve inverse kinematics problems. It gives us very good results [10]. This

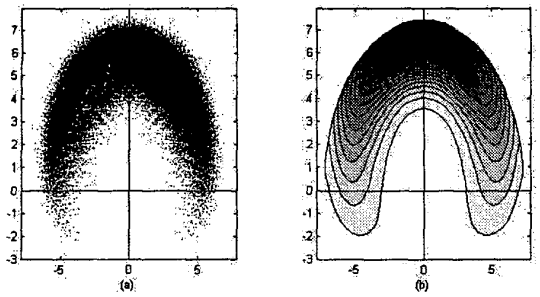

Figure 6: The workspace (a) and workspace density (b) of a 5-module manipulator

workspace generation algorithm is very efficient for manipulator workspaces which have a crescent shape.

6 Conclusion

In this paper, it was shown that the workspace density of a highly articulated manipulator can be found by solving a partial differential equation which has an explicit solution. The computational complexity of this algorithm is independent of the number of modules/DOF of a manipulator. In this sense, this algorithm is very suitable for highly articulated manipulators. To our knowledge, this is the only method in which computational complexity is independent of the number of modules/DOF. The effectiveness of this algorithm was verified by numerical simulations.

References

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

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

[3] G.S. Chirikjian, Y.F. Wang, "Conformational statistics of stiff macromolecules as solutions to PDEs on the rotation and motion Groups," *Physical Review E*, Vol. 62, No. 1, pp. 880-892, July 2000.

[4] G.S. Chirikjian, I. Ebert-Uphoff, "Numerical convolution on the Euclidean group with applications to workspace generation," *IEEE Transactions on Robotics and Automation*, pp. 123-136, Vol.14, No. 1, February 1998.

[5] I. Ebert-Uphoff, G.S. Chirikjian, "Efficient workspace generation for binary manipulators with many actuators," *J. of Robotic Systems*, Vol.12, No. 6, pp. 383-400, 1995.

[6] S.J. Kwon, Y. Youm, W.K. Chung, "General algorithm for automatic generation of the workspace for n-link planar redundant manipulators," *ASME Transactions*, Vol. 116, pp. 967-969, 1994.

[7] A.B. Kyatkin, G.S. Chirikjian, "Synthesis of binary manipulators using the Fourier transform on the Euclidean group," *ASME J. Mechanical Design*, pp. 9-14, Vol. 121, March 1999.

[8] J. Rastegar, P. Deravi, "Methods to determine workspace, its subspaces with different numbers of configurations and all the possible configurations of a manipulator," *Mechanism and Machine Theory*, Vol. 22, No. 4, pp. 343-350, 1987.

[9] R. Ricard, C. Gosselin, "On the determination of the workspace of complex planar robotic manipulators," *Proceedings ASME Mechanisms Conference*, Vol. DE-72, pp. 133-140, Sep. 1994.

[10] Y.F. Wang, *Applications of Diffusion Processes in Robotics, Optical Communications and Polymer Science*, Ph.D disserstation, Johns Hopkins University, 2001.