Alexander B. Kyatkin
Gregory S. Chirikjian

Department of Mechanical Engineering
Johns Hopkins University
Baltimore, Maryland, USA

Computation of Robot Configuration and Workspaces via the Fourier Transform on the Discrete-Motion Group

Abstract

We apply the Fourier transform on the discrete-motion group to the problem of computing the configuration-space obstacles of mobile robots which move among static obstacles, the problem of finding the workspace density of binary manipulators with many actuators, and the problem of determining workspace boundaries of manipulators with continuous-motion actuators. We develop and implement Fourier transforms for the discrete-motion group of the plane. These transforms allow us to apply fast Fourier transform methods to the computation of convolution-like integrals that arise in robot kinematics and motion planning. The results of the implementation are discussed for particular examples.

1. Introduction

In this paper, we address two problems in robotics which may be solved using similar mathematical tools: (1) the problem of computing configuration-space obstacles for mobile robots, and (2) the problem of calculating the workspaces of highly articulated serial and hybrid serial-parallel manipulators. In the context of problem 2, we focus on the problem of computing the workspace density function for discretely actuated m-link revolute manipulators and variable-geometry truss (vgt) manipulators. We also illustrate how these techniques are applied to the case of continuous-motion actuation.

All of these problems may be solved by using techniques from an area of mathematics called *noncommutative harmonic analysis* (Chirikjian and Kyatkin, 1999a). In particular, we use the Fourier transform on the Euclidean motion group (i.e., the group of translations and rotations of a rigid body) and a closely related group called the discrete-motion group. The Euclidean motion group (also called the special Euclidean group) is denoted as $SE(n)$, and can be viewed as the set of all $(n+1)\times(n+1)$ homogeneous transform matrices of the form

$$g = \begin{pmatrix} R & \mathbf{b} \\ \mathbf{0}^T & 1 \end{pmatrix},$$

where $R \in SO(n)$ (i.e., R is an $n \times n$ proper orthogonal matrix: $RR^T = 1$ and $\det(R) = +1$), $\mathbf{b} \in \mathbb{R}^n$, and the group operation is matrix multiplication of two homogeneous transform matrices.[1]

For the case of rigid-body motion in the plane ($SE(2)$), the homogeneous transforms take the particularly simple form

$$g(b_1, b_2, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta & b_1 \\ \sin\theta & \cos\theta & b_2 \\ 0 & 0 & 1 \end{pmatrix}.$$

The subgroup of $SE(2)$ where $\theta = 2\pi\, i/N$ for $i = 0, \ldots, N-1$ is called the *discrete-motion group* of the plane. While $SE(n)$ has been studied previously in various applications in physics and robotics (e.g., Vilenkin 1956; McCarthy 1991; Basavaraj and Duffy 1993; Murray, Li, and Sastry 1994; Park and Brockett 1994), and discrete subgroups of $SE(N)$ (where both rotation and translation are discretized) have been studied extensively in crystallography (e.g., Janssen 1973), the discrete-motion group defined above is far less familiar to the robotics community. In fact, the authors are aware of only one previous work using this group (Gauthier, Bornard, and Sibermann 1991).

* The word “on” is shorthand for “of functions on.”

The International Journal of Robotics Research
Vol. 18, No. 6, June 1999, pp. 601-615,

[1] Often throughout the text, motion-group elements are expressed simply as a translation-rotation pair, and written as $(R, \mathbf{b})$ or $(\mathbf{b}, R)$. In addition to $\mathbf{b}$, the symbols $\mathbf{x}$, $\mathbf{y}$, $\mathbf{a}$ and $\mathbf{r}$ are used for translations. Rotations are denoted as R and A.

We develop and apply harmonic analysis on the discrete-motion group. Robotics applications of harmonic analysis on $SE(N)$ have been addressed previously by the authors in several works in the context of discretely actuated manipulators (Ebert-Uphoff and Chirikjian 1995; Kyatkin and Chirikjian, 1999a). While we have devoted extensive effort to the concept of discretely actuated manipulators, it is actually quite an old idea. The concept can be traced back to work done in the Stanford robotics group more than a quarter of a century ago (Pieper 1968; Roth, Rastegar, and Scheinman 1973). Independently, research in the former Soviet Union along these lines was performed more than a decade ago (Koliskor 1986). Despite these efforts, we believe the concept did not catch on in the robotics community because two important tools did not exist: (1) a framework to handle the combinatorially explosive nature of the inverse-kinematics problem, and (2) a means for designing discretely actuated manipulators so that they could reach a finite set of specified frames in space.

The authors and their coworkers revived the concept of discretely actuated manipulators in the context of *binary* (two-state) actuation. In a series of papers, we have explored the design problem of selecting actuator stops so as to exactly reach a finite set of desired positions (Chirikjian 1994, 1995). We have also introduced the concept of workspace density together with algorithms to calculate this quantity (Ebert-Uphoff and Chirikjian 1995, 1996b). The workspace density of a binary manipulator (and each of its subunits) is an important quantity for performing inverse kinematics using the techniques of Ebert-Uphoff and Chirikjian (1996a). This technique is one of a number of techniques we have developed which circumvent the combinatorial explosive problems of binary manipulator inverse kinematics and trajectory planning. See the works of Ebert-Uphoff, Chirikjian, and Lees for other techniques (Ebert-Uphoff and Chirikjian 1996a; Lees and Chirikjian 1996a, 1996b, 1996c; Chirikjian 1997).

The workspace-density concept has greater importance than its application in binary manipulator inverse kinematics. It also provides a measure of how accurately any discretely actuated manipulator (including robots actuated with stepper motors) can reach any given position and orientation. The concept is also useful for manipulators with continuous actuation, since the range of each joint of a continuous-motion manipulator can be divided into discrete increments, and our methods can be easily applied. In another recent work, we explain how the propagation of error in serial and hybrid serial-parallel manipulators is formulated as an $SE(n)$-convolution of error density functions (Chirikjian and Kyatkin, 1999a). Given the wide variety of robotics applications mentioned above, we have therefore explored in great detail techniques from noncommutative harmonic analysis. Thus far the main problem we have addressed using these methods is the mathematical inverse problem of designing manipulators to have a desired workspace density (Chirikjian 1996; Kyatkin and Chirikjian, 1999a). We have also recognized that exactly the same techniques can be used to reformulate mathematical inverse problems in medical imaging in an elegant and computationally efficient way (Chirikjian and Kyatkin, 1999a).

The present work differs from our other works which use the Fourier transform on the Euclidean motion group, in that our goal here is not to synthesize a manipulator, but rather to generate the density for a given manipulator in an extremely efficient way. This builds on previous work (Ebert-Uphoff and Chrikjian 1995, 1996b), where the connection between manipulator workspaces, swept volumes, and convolution of functions on the Euclidean group has been explored. Here we define a group called the *discrete-motion group*, which approximates the Euclidean group. We use usual Abelian FFT techniques together with harmonic analysis on the discrete-motion group to approximate Euclidean group convolutions very efficiently. This not only has applications in robotics, but also in pattern matching (Kyatkin and Chirikjian, 1999b).

The advantage of using the discrete-motion group is that this allows us to apply fast Fourier transform (FFT) algorithms (Cooley and Tukey 1965; Elliott and Rao 1982). In practice, this reduces the time required from hours to minutes for the examples in this paper.

We develop and implement Fourier methods (using the FFT) on the discrete-motion group for the 2-D case, and apply them to the calculation of the configuration-space obstacles of mobile robots and the problem of computing the workspace density of binary vgt and revolute manipulators. This work is a greatly expanded version (with full mathematical detail) of ideas presented in an earlier work (Kyatkin and Chirikjian 1998a).

We begin by reviewing briefly these problems.

1.1. Computing the Configuration-Space Obstacles of Mobile Robots

The question of computing the configuration-space obstacles of a mobile robot has been considered in a number of papers (e.g., Lozano-Perez 1983; Newman and Branicky 1991; Guibas, Ramshaw, and Stolfi 1983; Kavraki 1995). Some of these algorithms (Lozano-Perez 1983) compute analytically the boundary of the regions of free configuration space for polygonal robots and obstacles. In this paper, we develop and implement a method that builds on the works of Kavraki (1995) and Guibas, Ramshaw, and Stolfi (1983), which may be applied to both polygonal and nonpolygonal shapes. We suggest a faster implementation of this method of computing configuration-space obstacles.

The mathematical formulation of the method is given below. We compute a function on the configuration space (the space of translations and rotations of the robot) which has nonzero values only in regions where the robot hits the obstacle. The magnitude of this "density function" is the ratio of the overlapping volume (area) of the robot to its total volume (area); i.e., it changes from 0 to 1 in the overlapping regions.

To calculate this value, we compute the integral

$$c(\mathbf{x}, A) = \frac{\int_{\mathbb{R}^n} f_1(\mathbf{y})\, f_2(A^{-1}(\mathbf{y}-\mathbf{x}))\, d^n y}{\int_{\mathbb{R}^n} f_2(\mathbf{y})\, d^n y}, \qquad (1)$$

where $f_{1,2}(\mathbf{x})$ are equal to 1 if the vector $\mathbf{x}$ is inside or on the boundary of the obstacles (robot), and zero otherwise; $n = 2, 3$ for 2-D (3-D) coordinate space; and $d^n y = dy_1 ... dy_n$ is the usual integration measure for $\mathbb{R}^n$. The function $c(\mathbf{x}, A)$ (where $\mathbf{x} \in \mathbb{R}^n$ and $A \in SO(n)$) is normalized to have a maximal value of 1; i.e., it is divided by the volume (area) of the robot. The geometrical meaning of this function is that it is zero when the obstacle and robot do not intersect, and has increasing positive value as the area of intersection increases.

To compute this integral directly, or simply to check pixel by pixel, that the obstacle $f_1(\mathbf{y})$ and the rotated and translated robot $f_2(A^{-1}(\mathbf{y}-\mathbf{x}))$ do not overlap, we need to perform $O(N\,N_r^2)$ computations, where N is the number of sampled orientations, and N_r is the number of sampled points in a bounded region of $\mathbb{R}^n$. For a large 3-D array of values, the computation of this integral by direct summation may be quite slow. The "overlap function" $c(\mathbf{x}, A)$, however, may be computed in $O(N\,N_r \log N_r)$ computations using Fourier methods on the discrete-motion group and FFT methods. This implementation is addressed in Sections 2 and 3. We discuss applications of this method to mobile-robot configuration-space generation in detail in Section 4.1.

1.2. Generating Workspaces of Manipulators

We also apply the Fourier transform on the discrete-motion group to the workspace generation of planar manipulators, with a particular emphasis on binary manipulators; for a review of binary manipulators, see the works of Chirikjian and Ebert-Uphoff (Chirikjian 1994, 1995; Ebert-Uphoff and Chirikjian 1995).

Binary manipulators are robot arms with discrete-state actuators. The *workspace density*, defined as the number of reachable frames per unit volume of the Euclidean motion group *SE*(n) (Chirikjian and Ebert-Uphoff 1998; Ebert-Uphoff and Chirikjian 1996b) determines how accurately a position and orientation can be reached. The workspace-density information is important for the kinematic design of manipulators and for planning the motions of discretely actuated manipulator arms (Ebert-Uphoff and Chirikjian 1996a). In Figure 1, we depict a six-module binary vgt manipulator and gray scale of the $\theta = 0$ (zero end-effector orientation angle) slice of the workspace density. The workspace density of m-link discretely actuated revolute manipulators like the one shown in Figure 2 is treated in exactly the same way. An alternative visualization of workspace density is to view level curves of each slice of the workspace for fixed θ.

To compute the workspace-density function using brute force and iterating is computationally unrealistic for a large number of actuators m, because it requires K^m evaluations of

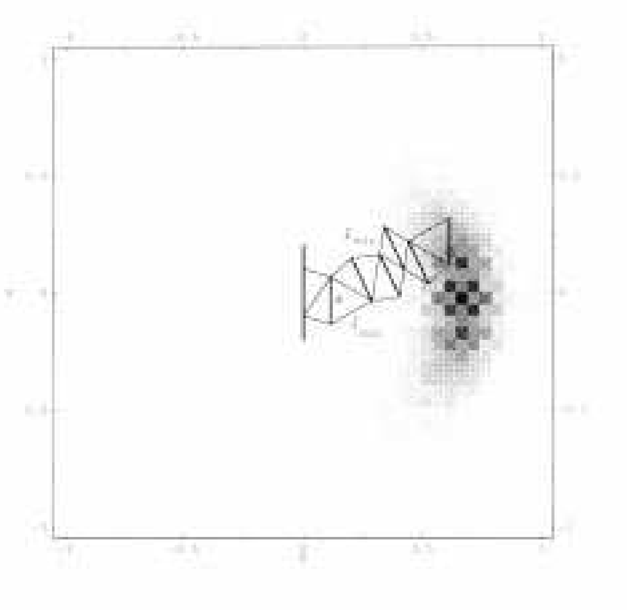


Fig. 1. A six-module manipulator and its workspace density for a $\theta = 0$ orientation angle.

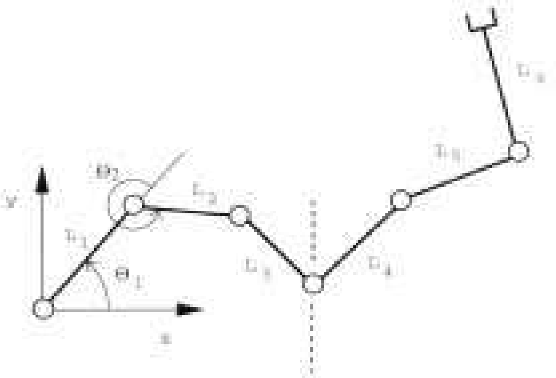


Fig. 2. Illustration of a six-link discretely actuated revolute planar manipulator.

the kinematic equations relating actuator state to the resulting end frame for a manipulator with m actuated modules, each with K states. However, as has been shown (Chirikjian and Ebert-Uphoff 1998; Ebert-Uphoff and Chirikjian 1996b), partitioning the manipulator into segments, computing the workspace-density function for each segment, and computing the workspace-density function of the whole manipulator as a convolution of those for each segment is an efficient way to solve this problem. Using the convolution approach allows us to reduce the exponential growth in m to linear growth in m for inequivalent segments and to $O(\log m)$ for identical segments.

We note that the concept of workspace density, while initially motivated by binary manipulators, is also applicable to manipulators with continuous motion. In the context of continuous actuation, a manipulator's workspace-density function is calculated by discretizing the range of each joint, and calculating it in exactly the same way as for a binary manipulator. The density function is thus an easily computable measure of redundancy of the arm in the sense that it immediately yields the number of configurations of the arm that can reach a given voxel in $SE(n)$. If one is uninterested in this density, only the location of $SE(n)$ voxels where the density transitions to zero can be stored, thus yielding the workspace boundary in position and orientation. The importance of calculating manipulator workspaces has been explained elsewhere in the literature (e.g., Kwon, Youm, and Chung 1994; Ceccarelli and Vinciguerra 1995).

The key to the formulation presented here is that the workspace density of a manipulator which has double the number of identical modules as one with workspace density $f(g)$ may be generated as a convolution on the motion group

$$F(g) = (f * f)(g) = \int_{SE(n)} f(h)\, f(h^{-1} \circ g) d\mu(h), \quad (2)$$

where $d\mu(h)$ is a left- and right-invariant integration measure on $SE(n)$, i.e., $d\mu(g \circ h) = d\mu(h \circ g) = d\mu(g)$.

Throughout this paper, we will interchange notations for this convolution. For instance, any function on the Euclidean motion group $SE(n)$ can be written as $f(g) = f(\mathbf{x}, A)$, where $\mathbf{x} \in \mathbb{R}^n$ and $A \in SO(n)$. Using this notation, the above convolution integral is written as

$$F(\mathbf{x}, A) = \int_{SO(n)} \int_{\mathbb{R}^n} f(\mathbf{y}, R)\, f(R^{-1}(\mathbf{x} - \mathbf{y}), R^{-1}A)\, d^n y\, dR,$$

where dR is the normalized left-right-invariant integration measure on $SO(n)$. While the idea of such integration measures is relatively new in the robotics literature (Basavaraj and Duffy 1993; Murray, Li, and Sastry 1994; Park and Brockett 1994), such measures have been known in the mathematical physics literature for decades (Vilenkin 1956), and so we do not discuss them here.

In this paper, we concentrate on the planar case, where this convolution integral is written in its most concrete form as

$$F(x_1, x_2; \theta) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\pi}^{\pi} f(y_1, y_2; \alpha)$$
$$f((x_1 - y_1)\cos\alpha + (x_2 - y_2)\sin\alpha,$$
$$-(x_1 - y_1)\sin\alpha + (x_2 - y_2)\cos\alpha; \theta - \alpha)\, dy_1\, dy_2\, d\theta.$$

This integral[2] may be computed efficiently using the Fourier transform on the motion group (in analogy with the use of the usual Fourier transform to compute convolutions of functions on the real line). This problem is addressed in Sections 2, 3, and 4.2. The efficiency of the Fourier transform is discussed in Section 3.

2. Fourier Transform on the Discrete Motion Group

To get a simple expression for the convolution integral in Fourier space, we have to use a generalized Fourier transform with the property that

$$\mathcal{F}(f_1 * f_2) = \mathcal{F}(f_2)\mathcal{F}(f_1).$$

A well-developed theory for such generalizations of the Fourier transform exists. It is called noncommutative harmonic analysis. A key element of this theory is the enumeration of linear operators, U, which have the property

$$U(g_1)\, U(g_2) = U(g_1 \circ g_2), \quad (3)$$

where $g_{1,2}$ are group elements of a group G, and the operator product may be understood as a matrix product (of, in general, infinite dimensional matrices). This homomorphism property allows one to reduce the convolution integral (eq. (2)) to a matrix-product equation in Fourier space. The property described by eq. (3) is just part of the definition of a group representation (Sugiura 1990), which tells us that we have to use these operators to get Fourier transforms with the convolution property. The operators U can be thought of as generalizations of the complex exponentials used in usual Fourier analysis.

We have to use a complete and orthonormal set of Fourier matrix elements (to make sure that the application of the direct and inverse Fourier transform reproduces the original function, and, again, to reduce the convolution integral to a matrix-product equation). To generate the complete and orthonormal set of matrix elements, we have to use *irreducible* and *unitary* representations. A representation is irreducible if no nontrivial subspace exists in the space of functions where the operators of the representation act (Sugiura 1990; Gurarie 1992).

[2] The factor of $1/(2\pi)^2$ is a matter of choice which is consistent with the mathematics literature (e.g., Sugiura 1990).

The representation is a unitary representation, if it does not change the inner product on that space (Sugiura 1990; Gurarie 1992),

$$(U\cdot f, U\cdot h) = (f, h),$$

where f, h are the functions in the space (called the representation space, and denoted as V), where the operators act. The inner product $(\cdot,\cdot)$ acts on functions in the space V, and the actual form of the linear operators U (computed as matrices) depends on which basis for V is chosen.

For example, the functions $\exp(i\, m\theta)$ in the usual Fourier series on the unit circle are the irreducible unitary representations of $SO(2)$, enumerated by the integer m. In this case, the representation matrices are one dimensional.

2.1. Irreducible Unitary Representations of the Discrete-Motion Group

In this subsection, we define matrix elements of the irreducible unitary representations (IURs) of the discrete-motion group $G = \mathbb{R}^2 \rhd C_N$, where C_N is the N-element finite subgroup of $SO(2)$ (i.e., the group of rotational symmetry operations of a regular N-gon), and the notation $\rhd$ means a semidirect product (which in the present context means nothing more than that group elements are expressed as homogeneous transforms). The continuous-motion group $SE(2) = \mathbb{R}^2 \rhd SO(2)$ is considered by Vilenkin (1956), Orihara (1961), Talman (1968), Chirikjian (1996) and Kyatkin and Chirikjian (1999a). Fourier analysis and FFTs for certain classes of finite groups are considered by Rockmore (1994), and sampling theorems for compact groups are presented by Maslen (1993). However, the discrete-motion group is neither finite nor compact, and so these methods are not directly applicable here.

The IURs $U(\mathbf{r}, A)$, for $(\mathbf{r}, A) \in SE(2)$, act on functions $f(\mathbf{u}) \in L^2(S^1)$, where S^1 is the unit circle, and $\mathbf{u} = (\cos\theta, \sin\theta)^{\mathrm{T}}$ is a vector to a point on the unit circle. The inner product of functions on the circle is

$$(f\cdot h) = \frac{1}{2\pi}\int_{S^1} \overline{f(\theta)}\, h(\theta)\, d\theta,$$

and θ is the angle measured counterclockwise from the x-axis. The IUR operators are defined by the expression

$$\begin{aligned}(U(\mathbf{r}, A; p)\, f)(\mathbf{u}) &= e^{ip\mathbf{u}\cdot\mathbf{r}}\, f(A^{-1}\mathbf{u}) \\ &= e^{i\mathbf{p}\cdot\mathbf{r}}\, f(A^{-1}\cdot(\mathbf{p}/p)),\end{aligned} \qquad (4)$$

where $A \in SO(2)$, $p \in \mathbb{R}^+$, and $\mathbf{p} = p\mathbf{u}$ is the vector to arbitrary points in the dual (frequency) space of $\mathbb{R}^2$ (p is its magnitude and $\mathbf{u}$ is its direction).

We define an orthonormal set of pulse functions $f_{N,n}(\mathbf{u})$ on S^1 by subdividing the circle into identical segments, F_n, and choosing the pulse functions to satisfy the orthonormality relations

$$\frac{1}{2\pi}\int_{S^1} f_{N,n}(\mathbf{u})\, f_{N,m}(\mathbf{u})\, d\theta = \delta_{nm}.$$

We may choose the orthonormal functions as

$$f_{N,n}(\mathbf{p}) = \begin{cases} (N)^{1/2} & \text{if } \mathbf{u} \in F_n \\ 0 & \text{otherwise,} \end{cases}$$

where $n = 0, ..., N-1$ enumerates different segments. We denote these pulse functions as δ-like functions $f_{N,n}(\mathbf{u}) = (1/N)^{1/2}\, \delta_N(\mathbf{u}, \mathbf{u_n})$, where $\mathbf{u}_n$ is the vector to the center of the F_n segment.

The matrix elements of U in this orthonormal set of functions are

$$U_{mn}(A, \mathbf{r}; p) = \frac{1}{2\pi}\int_{S^1} f_{N,m}(\mathbf{u})\, e^{i\,p\,\mathbf{u}\cdot\mathbf{r}}\, f_{N,n}(A^{-1}\mathbf{u})\, d\theta. \qquad (5)$$

Using the "delta-function" notations, this may be written as

$$U_{mn}(A, \mathbf{r}; p) = \frac{1}{2\pi\, N}\int_{S^1} \delta_N(\mathbf{u}, \mathbf{u}_m)\, e^{i\,p\,\mathbf{u}\cdot\mathbf{r}}\, \delta_N(A^{-1}\mathbf{u}, \mathbf{u}_n)\, d\theta.$$

This integral may be approximated as

$$U_{mn}(A, \mathbf{r}; p) \approx 1/N\, e^{i\,p\,\mathbf{u}_m\cdot\mathbf{r}}\, \delta_N(A^{-1}\mathbf{u}_m, \mathbf{u}_n). \qquad (6)$$

We approximate delta functions as

$$\begin{aligned} 1/N\, \delta_N(A^{-1}\mathbf{u}_m, \mathbf{u}_n) &= \delta_{A^{-1}\mathbf{u}_m, \mathbf{u}_n} \\ &= \begin{cases} 1 & if\ A^{-1}\mathbf{u}_m = \mathbf{u}_n \\ 0 & \text{otherwise,} \end{cases}\end{aligned}$$

which means that we restrict rotations to the rotations A_j from the finite subgroup C_N of $SO(2)$, and $A_j^{-1}\mathbf{u}_m = \mathbf{u}_{m-j} = \mathbf{u}_n$.

Thus, the matrix elements of the irreducible unitary representations of the "discrete" motion subgroup $G = \mathbb{R}^2 \rhd C_N$ are given as

$$U_{mn}(A_j, \mathbf{r}; p) = e^{i\,p\,\mathbf{u}_m\cdot\mathbf{r}}\, \delta_{A_j^{-1}\mathbf{u}_m, \mathbf{u}_n}, \qquad (7)$$

where $\delta_{A_j^{-1}\mathbf{u}_m, \mathbf{u}_n} = \delta_{m-j,n}$ in this case.

2.2. Fourier Transforms on the Discrete-Motion Group

The matrix elements of eq. (7) are exact expressions for the matrix elements of the unitary representations of the discrete-motion group. The set of matrix elements in eq. (7) is, however, incomplete. This means that the direct and inverse Fourier transforms, defined using these matrix elements, would reproduce the original function with $O(1/N)$ error; i.e.,

$$\mathcal{F}^{-1}(\mathcal{F}(f(A_i, \mathbf{r})) = f(A_i, \mathbf{r})\,(1 + O(\frac{1}{N})).$$

The reason for this is that summing through all possible segments cannot replace integration over all possible angles on the circle. It is clear that an additional continuous parameter that enumerates possible angles inside each segment on the circle is required to enumerate the complete set of the matrix elements.

Thus, the matrix elements are modified as

$$U_{mn}(A_j, \mathbf{r}; p, \phi) = e^{i\,p\,\mathbf{u}_m^{\phi}\cdot\mathbf{r}}\,\delta_{A_j^{-1}\mathbf{u}_m, \mathbf{u}_n}, \tag{8}$$

where $\mathbf{u}_k^{\phi}$ denotes the vector to the angle θ on the unit circle on the interval $F_k = [2\pi\,k/N\,, 2\pi\,(k+1)/N]$, $k = 0, ..., N-1$ (ϕ measures the angle within this segment).

The completeness relation

$$\sum_m \sum_n \int_0^{\infty} \int_0^{2\pi/N} \overline{U_{mn}(A_i, \mathbf{r}_1; p, \phi)}\, U_{mn}(A_j, \mathbf{r}_2; \\ p, \phi)\, p\, dp\, d\phi = (2\pi)^2\, \delta^2(\mathbf{r}_1 - \mathbf{r}_2)\, \delta_{A_i, A_j} \tag{9}$$

is exact, because the integration is now over the whole space of $\mathbf{p}$ values. This may be proven using the integral representation of the δ function

$$\frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} e^{i\,\mathbf{p}\cdot\mathbf{r}}\, d^2p = \delta^2(\mathbf{r}).$$

The orthogonality relation is written as

$$\sum_{i=0}^{N-1} \int_{\mathbb{R}^n} \overline{U_{mn}(A_i, \mathbf{r}; p, \phi)}\, U_{m'n'}(A_i, \mathbf{r}; p', \phi')\, d^2r = \\ (2\pi)^2\, \frac{\delta(p - p')}{p}\, \delta_{m,m'}\, \delta_{n,n'}\, \delta(\phi - \phi'). \tag{10}$$

The direct Fourier transform is defined as

$$\hat{f}_{mn}(p, \phi) = \sum_{i=0}^{N-1} \int_{\mathbb{R}^2} f(A_i, \mathbf{r})\, U_{mn}^{-1}(A_i, \mathbf{r}; p, \phi)\, d^2r. \tag{11}$$

The vector $\mathbf{u}_m^{\phi}$, which is inside the segment F_m, may be received by the rotation A_m (which transforms F_0 to F_m) from $\mathbf{u}_0^{\phi}$, $\mathbf{u}_m^{\phi} = A_m, \mathbf{u}_0^{\phi}$. The parameter ϕ denotes the position inside the segment F_0.

The inverse Fourier transform is

$$\mathcal{F}^{-1}(\hat{f}) = \frac{1}{4\pi^2} \sum_m \sum_n \int_0^{\infty} \int_0^{2\pi/N} \\ \hat{f}_{mn}(p, \phi)\, U_{nm}(A_i, \mathbf{r}; p, \phi)\, p\, dp\, d\phi. \tag{12}$$

We note that this result is in agreement with Gauthier, Bornard, and Sibermann (1991), and is a result of the completeness relation (eq. (9)) and the orthogonality relation (eq. (10)).

We define convolution on the discrete-motion group as

$$F(\mathbf{r}, A_j) = \frac{1}{2\pi N} \sum_{i=0}^{N-1} \int_{\mathbb{R}^2} f_1(\mathbf{a}, A_i) \\ f_2(A_i^{-1}(\mathbf{r} - \mathbf{a}), A_{j-i}) d^2a. \tag{13}$$

The normalization factor corresponds in the limit $N \to \infty$ to the normalization of the convolution on the continuous (Euclidean) motion group, which is discussed by Kyatkin and Chirikjian (forthcoming).

The Fourier transform of the convolution of functions on the discrete-motion group is just a product of the Fourier matrices with the corresponding normalization

$$\hat{F}_{mn}^{\phi}(p) = \frac{A}{2\pi N} \sum_{k=0}^{N-1} (\hat{f}_2)_{mk}^{\phi}(p)\, (\hat{f}_1)_{kn}^{\phi}(p), \tag{14}$$

where A is the area of the compact region of $\mathbb{R}^2$ where the FFT is computed. The functions $f_i(\cdot)$ must have support inside the area A, and they are considered periodic outside of this region. The area factor arises because the discrete Fourier transform may be obtained as an approximation of the continuous case using

$$r_1 \to \frac{L}{N_r} i; \quad p_1 \to \frac{2\pi}{L} i,$$

(and analogously for the second component). Here L is the length of the compact region in the x- (y-) direction. While the factor L is canceled out of equations if we apply direct and inverse discrete-motion-group Fourier transforms to the function, it appears in the convolution, defined in eq. (13).

3. Efficiency of Computation of Convolution Integrals using the Fourier Transform

In this section, we show that using the Fourier transform on the discrete-motion group is a fast method for computing convolution integrals on the discrete-motion group (we assume below that the finite-rotation group has N elements). Particularly, we show that the convolution of a function $f(\mathbf{r}, A_i)$ sampled at $N_g = N \cdot N_r$ points (N_r is the number of samples in an $\mathbb{R}^2$ region) may be performed in $O(N_g\ \log N_r) + O(N_g\, N)$ operations, instead of the $O(N_g^2)$ computations required in direct coordinate-space integration using the “plain” integration in eq. (13). The structure of matrix elements in eq. (7) allows one to apply fast Fourier methods and reduce the amount of computations (without the application of FFT, the amount of computations using the Fourier transform method is $O(N_g^2/N)$).

First, we estimate the amount of computations to perform the direct and inverse Fourier transforms of $f(g)$. We restrict p values to a finite interval and sample it at N_p points, and sample the ϕ values at N_ϕ points. We also assume that the total number of harmonics $N_p N_\phi N^2 = N_g = N N_r$.

Let us consider the direct Fourier transform in eq. (11). Each term i (for fixed A_i) gives one nonzero term in each row and column of the Fourier matrix $\hat{f}^{\phi}_{mn}(p)$ (because only one element in each row and column of $U^{-1}_{mn}(g;\, p, \phi)$ is nonzero). For each fixed i, we may compute the usual FFT of $f(\mathbf{r};\, A_i)$, which may be computed in $O(N_r \log(N_r))$ operations. The Fourier transform elements calculated by the FFT are computed on a square grid of $\mathbf{p}$ values. We, however, may interpolate the Fourier elements computed on the grid to Fourier elements computed in polar coordinates. The radial part p is determined by the length of $\mathbf{p}$, the angular part determines the indices m and ϕ, and the index n is determined uniquely for given A_i. We linearly interpolate values on a square grid to values on a polar grid. Such an interpolation may be performed in $O(N_r)$ computations. Each term i in eq. (11) may be computed in $O(N_r \log(N_r))$ computations. The whole Fourier matrix may be computed in $O(N\, N_r \log(N_r))$ computations.

Again, one element from each row and column is used in the computation of the inverse Fourier transform for each rotation element A_i. After inverse interpolation to Cartesian coordinates (which may be done in $O(N_r)$ computations), the inverse Fourier integration may be performed in $O(N_r \log(N_r))$ for each of the N nonzero matrix elements of U using the FFT. Thus, in $O(N\, N_r \log(N_r))$ computations, we reproduce the function for all A_i.

The matrix product of $\hat{f}^{\phi}_{mn}(p)$ may be computed directly in $O(N^3)$ computations [3] for each value of p and parameters ϕ. This means that the convolution (which is a matrix product of Fourier matrices) may be performed in $O(N^3 N_p\, N_\phi) = O(N N_g)$ computations.

Therefore, the convolution of functions on the discrete-motion group may be performed in $O(N_g\ \log N_r) + O(N_g\, N)$ using Fourier methods on the discrete-motion group and the usual FFT, without assuming any special matrix-multiplication technique.

It may be shown that without the application of the FFT the convolution may be performed using the Fourier transform in $O(N_g^2/N)$, which is still faster than evaluating the direct integration.

4. Applications of the Fourier Transform on the Discrete-Motion Group

In this section, three example applications of these techniques in robotics are demonstrated. In Section 4.1, a configuration-space obstacle of a mobile robot is calculated. In Section 4.2, the workspace-density function of a variable-geometry-truss-manipulator is calculated. In Section 4.3, the workspace density of an m-link revolute manipulator is calculated.

4.1. Configuration-Space Obstacles of a Mobile Robot

Here we apply the numerically implemented Fourier transform on the discrete-motion group of the plane to the computation of the free configuration space of a rigid mobile robot moving among static 2-D obstacles. To find the configuration-space obstacles, we compute the integral (eq. (1)) using the above-described Fourier methods for the discrete-motion group. Because the obstacles and the robot are functions only of Cartesian position (i.e., to use eq. (1), we need to compute the Fourier transform of $f_2(\mathbf{x})$ only for one fixed orientation), the direct Fourier transform is performed faster than for an arbitrary function on the motion group.

A function of position may be considered as a function on the discrete-motion group which does not depend on the orientation; i.e., $f(\mathbf{x}, A_i) = f(\mathbf{x})$. The "overlap" function of eq. (1) may be formally written as the integral

$$c(\mathbf{x}, A_j) = \frac{1}{N} \frac{\displaystyle\sum_{i=0}^{N-1} \int_{\mathbb{R}^2} f_1(\mathbf{y}, A_i)\, f_2(A_j^{-1}(\mathbf{y}-\mathbf{x}), A_j^{-1} A_i)\, d^2y}{\displaystyle\int_{\mathbb{R}^2} f_2(\mathbf{y})\, d^2y}. \tag{15}$$

Because the functions are real the integral in the numerator may be written as

$$\frac{1}{N} \sum_{i=0}^{N-1} \int_{\mathbb{R}^2} \overline{f_1(\mathbf{y}, A_i)}\, f_2(A_j^{-1}(\mathbf{y}-\mathbf{x}), A_j^{-1} A_i)\, d^2y$$

$$= \int_{G_N} \overline{f_1(h)}\, f_2(g^{-1} h)\, d\mu(h),$$

where we denote integration $d\mu(h)$ over the discrete-motion group, G_N, to mean integration with respect to $\mathbb{R}^2$ and the summation through the A_i, and the group elements are of the form $g = (\mathbf{x}, A_j)$. Using the orthogonality properties of the Fourier matrix elements of eq. (10), this integral may be written as

[3] Estimates as fast as $O(N^{2.38})$ have been reported (Coppersmith and Winograd 1987; Rockmore 1994), which, depending on how they are implemented, have the potential to increase speed further.

$$\frac{1}{4\pi^2 N}\sum_q\sum_n\int_0^\infty\int_\phi\sum_m\left(\overline{\hat{f}_{1mn}}\,\hat{f}_{2mq}\right)$$
$$U_{qn}(g^{-1};\,p,\phi)\,p\,dp\,d\phi \tag{16}$$
$$=\frac{1}{4\pi^2 N}\sum_q\sum_n\int_0^\infty\int_\phi\sum_m\left(\overline{\hat{f}_{2mq}}\,\hat{f}_{1mn}\right)$$
$$U_{nq}(g;\,p,\phi)\,p\,dp\,d\phi,$$

where integration with respect to ϕ is integration on the circle in the interval $F_q = [2\pi\, q/N\,, 2\pi\,(q+1)/N\,]$. For the second expression, we used the unitarity of the matrix elements and the fact that the expression is real (i.e., we take a complex conjugate of the integral). The matrices $(\hat{f}_{1,2})_{mn}$ are the Fourier transforms of the functions $f_{1,2}(\mathbf{x}, A_i)$.

Because functions $f_{1,2}(\mathbf{x}, A_i) = f_{1,2}(\mathbf{x})$ do not depend on the orientations A_i, matrix elements of the Fourier transforms in the same column are the same; i.e.,

$$(\hat{f}_{1,2})_{mn} = (\hat{f}_{1,2})_{qn}$$

for any m, q. This may be observed from the expression

$$U(g^{-1};\,p,\phi)_{mn} \;=\; e^{-i\,p\,\mathbf{u}_n^{\phi}\cdot\mathbf{r}}\;\delta_{A_i^{-1}\mathbf{u}_n,\mathbf{u}_m}$$

(the exponent depends only on the n-index), the definition of the direct transform (eq. (11)), and the fact that the functions do not depend on the orientation. Thus, we compute a row of the Fourier matrix for a particular orientation (for example, $A_0 = \mathbf{1}$),

$$(\hat{f}_{1,2})_n = (\hat{f}_{1,2})_{nn}.$$

This may be done using the 2-D FFT for the functions $f_{1,2}(\mathbf{x})$ and interpolating the Fourier values to points on a polar coordinate grid. This requires $O(N_r\,\log(N_r))$ computations. Thus, the "overlap" function, eq. (1), is written as

$$c(\mathbf{x}, A_j) \;=\; \frac{C}{4\pi^2}\;\frac{\sum_q\sum_n\int_0^\infty\int_\phi\left(\overline{\hat{f}_{2q}}\,\hat{f}_{1n}\right)U_{nq}(g;\,p,\phi)\,p\,dp\,d\phi}{\int_{\mathbb{R}^2} f_2(\mathbf{y})\,d^2y}, \tag{17}$$

where $C = \frac{1}{N}$. The product of the column $\overline{\hat{f}_2}$ and the row $\hat{f}_1$ may be performed in $O(N_g = N_r\,N)$ computations, and the inverse transform may be performed in $(N_g\,\log(N_r))$ computations. The normalization of the function $f_2(\mathbf{x})$ may be computed by direct integration in $O(N_r)$. Thus, the inverse transform is the largest time-consuming computation. We note that the direct Fourier transform is performed only for one orientation. This is N times faster (for discrete-rotation group C_N) than to perform the FFT for each orientation as is done by Kavraki (1995), but the inverse transform is of the same order of computations. Thus, as N becomes large, there is a built-in factor of two speed increase using our method. We also mention that inverse problems, i.e., problems of finding the allowed shape of the robot for the given desired configuration space and the shape of the obstacles, may be solved using Fourier methods on the motion group, because it is reduced to a functional matrix equation (regularization methods for the solution of singular functional matrix equations are discussed by Kyatkin and Chirikjian (1998b)).

When we discretize the coordinate regions and use the FFT to compute eq. (1), the coefficient C in eq. (17) must be defined as $C = A/N$, where A is the area of the compact region of $\mathbf{x}$ values where the FFT is computed (the functions must have a support inside the area A; the functions are considered periodic outside of this region). The area factor arises because the discrete Fourier transform is an approximation of the continuous case when using the transformation

$$x_1 \to \frac{L}{N_r}i;\quad p_1 \to \frac{2\pi}{L}i$$

(and analogously for the second component). Here L is the length of the compact region in the x- (y-) direction. While the factor L is canceled out of equations if we apply the direct and inverse discrete Fourier transform to the function, it appears in the convolution-like integral (eq. (1)).

Thus, we compute the 2-D FFT of $f_{1,2}(\mathbf{x})$, interpolate the Fourier elements to a polar grid, arrange them into the Fourier column and row, multiply column and row $\hat{f}_{mn} = \overline{\hat{f}_{2m}}\,\hat{f}_{1n}$, and take the inverse Fourier transform (the corresponding elements $m = n - i$ from the Fourier matrix $\hat{f}_{mn}$ must be taken for each orientation A_i, and interpolated back to a Cartesian grid to take the 2-D inverse FFT).

We implemented the computation of eq. (17) using the FFT in the C programming language. Time to compute $c(\mathbf{x}, A_j)$ was 30 sec (on a 250-MHz SGI workstation) for a 256 × 256 square grid in $\mathbb{R}^2$ for the C_{10} group ($N = 10$, and we subdivide each segment into $N_\phi = 20$ subsegments).

Because for small values of $c(\mathbf{x}, A_j)$ the function exhibits oscillations (due to finite discretization of the integration area in the Fourier transform), we depict the boundary of the configuration space, defined by the contour line where $c(\mathbf{x}, A_j) = \epsilon$. The smallest possible choice of ϵ in our examples was in the region $0.005 - 0.035$. To increase accuracy, we used the following method. We increased the value of $f_2(\mathbf{x})$ in the region near the border of the robot. For a convex robot this may be done by scaling down the robot and increasing the value of $f_2(\mathbf{x})$ between the scaled boundary of the robot and the original boundary.

While changing the function values in this "rim" region does not change the shape and size of the robot, the overlap of the robot and obstacle is effectively made more sensitive. If the overlap is completely in the "rim" of the increased values near the boundary, the following estimate is valid for the intersection area ϵ' for given value of ϵ:

$$\epsilon' = \epsilon/q,$$

where $q = \frac{V}{V-1/k^2(V-1)}$ (the function f_2 is equal to 1 in the area scaled down by the factor k, and increased to the value V in the "rim" near the boundary of the robot). ϵ' is a more accurate estimate of eq. (1) than is obtained by using the Fourier technique directly.

In addition, we may increase the size of the robot and depict the configuration space for the scaled robot. The area between the two configuration boundaries (of the scaled robot and the original robot) is the "near-collision" area of the robot and the obstacle. This is an important region for motion planning. We note that the direct method (i.e., direct integration of eq. (1) or a direct check if the robot and the obstacle overlap) can be used to find a precise boundary in the "near-collision" area. This may be performed in $O(N_g^2\, A/A_{tot})$ computations, where A/A_{tot} is the ratio of the "near-collision" region to the total region of the coordinate space. Because this ratio is generally small, this gives very considerable (hundreds of times) savings in computations over using the direct method for all configurations.

We note that this method may not give good results if a very narrow and long object is attached to the robot (i.e., when the intersection area is not sensitive to the overlap). However, the value of $f_2(\mathbf{x})$ may be increased in these regions, and this method may be useful for some of these shapes.

As an example, we depict slices of the configuration space in Figures 3 and 4. The obstacle has a polygonal shape.[4] The robot has an elliptic shape. The size and orientation of the robot are depicted in the lower-right corner on the figures. The solid line outside the obstacle depicts the boundary of the configuration space ($\epsilon \approx 0.030 - 0.035$). We increase the value of the function f_2 to 10 in the rim of the robot depicted in Figure 3. In this example, the rim is the region between the original boundary of the robot and the boundary scaled down by a factor $k = 1.1$. We also depict in the pictures the boundary (dashed line) that describes the configuration-space obstacles for the robot when the robot is located completely inside the walls of the obstacle ($g(\mathbf{x}, A) \approx 0.95 - 0.97$).

The figures also depict the corresponding boundaries for a scaled robot (enlarged by a factor of 1.27). Inside this region is where the exact position of the configuration-space boundary may be found by direct integration.

4.2. The Workspace Density of Two-Dimensional Binary Manipulators

First, we discuss briefly implementation of the Fourier transform of functions on the discrete-motion group for this case.

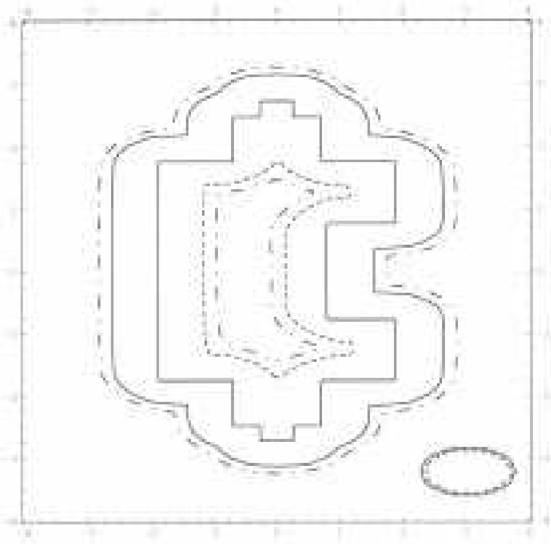


Fig. 3. A configuration-space obstacle for the $\theta = 0$ orientation of the robot (as shown in the lower-right corner).

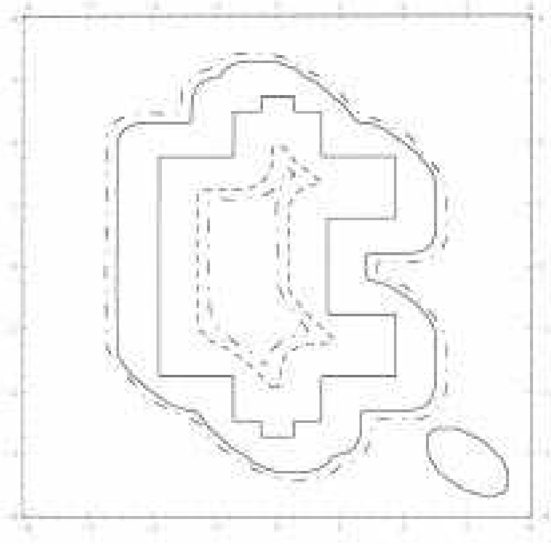


Fig. 4. The configuration-space obstacle for the $\theta = 4\pi/5$ orientation of the robot.

[4] The Fourier method is applicable both in the cases of polygonal and "smooth" obstacles. The results are better for "smoother" obstacles; i.e., the value of ϵ may be chosen smaller.

The matrix elements of the Fourier transform in eq. (7) for the rotation subgroup C_N may be written as

$$U_{mn}(A_j, \mathbf{r}; p, \phi) = e^{i\, p\, \mathbf{u}_m^{\phi} \cdot \mathbf{r}}\, \delta_{m-j,n},$$

and $\mathbf{u}_m^{\phi} \cdot \mathbf{r} = r \cos(2\pi/N\, m + 2\pi/(N N_\phi)\, \phi - \theta)$, where $m = 0, ..., N-1$; discrete values of ϕ are $\phi = 0, ..., N_\phi - 1$ (N_ϕ is a number of sample points on the interval $[2\pi/Nm, 2\pi/N\,(m+1)]$); and θ is the polar angle of $\mathbf{r}$.

The direct Fourier transform is given by eq. (11). Each term in the sum $i = 0, ..., N-1$ through the elements of the rotation group C_N gives one element of the Fourier transform matrix in each row and each column, and summing through all $i = 0, ..., N-1$ gives the whole matrix. For each term i in eq. (11), we restrict integration with respect to $\mathbf{r}$ to a compact region, sampled on the $N_r \times N_r$ square grid.

Since we want to use the advantages of the fast Fourier transform, we compute the two-dimensional FFT (for each i) and interpolate the value of the Fourier transform on the $N_r \times N_r$ square grid of $\mathbf{p}$ values to polar coordinates. The radial coordinate is $p = |\mathbf{p}|$, and the polar angles are determined by the values of m and ϕ (the value of n is determined from m and the index of rotation i). Thus, we must compute N FFTs to obtain the whole Fourier matrix in the case of N discrete rotations.

In the inverse Fourier transform, we have to interpolate back the Fourier parameter p from the polar coordinate to values of $\mathbf{p}$ on the square grid (we take one element from each row and column of the Fourier matrix for each rotation element i), and compute the two-dimensional inverse FFT for each rotation i. Again, the whole function $f(g)$ is reproduced in N FFTs.

The Fourier transform of convolution (eq. (13)) is defined by eq. (14).

We implemented the two-dimensional Fourier transform on the discrete-motion group in the C programming language.

We note that already for the C_{10} group, the quadratic deviation of the convolved functions on the discrete-motion group from the convolved functions on the continuous group is small. We calculated a 3.7% error in the quadratic deviation for the example below, where we defined a quadratic deviation of $f(g)$ from $f'(g)$ as

$$r = \frac{\sum_{i=0}^{N-1} \int_{\mathbb{R}^2} |f(\mathbf{x}, A_i) - f'(\mathbf{x}, A_i)|^2 d^2x}{\sum_{i=0}^{N-1} \int_{\mathbb{R}^2} |f(\mathbf{x}, A_i)|^2 d^2x}.$$

Thus, the Fourier transform on the discrete-motion group is a fast way to convolve functions (and it gives a good approximation to convolved functions on the continuous group). We calculate below the workspace-density function of a 12-module binary vgt manipulator,[5] calculated as the convolution of a 6-module manipulator workspace density with itself. The workspace density of the six-module manipulator, computed by brute force,[6] is depicted in Figure 5a (for the manipulator parameters $l_{\min} = 0.12$, $l_{\max} = 0.2$, $s = 0.2$, see Fig. 1). The convolved workspace density is depicted in Figure 5b. It describes the workspace of the 12-module manipulator. Time to compute the convolution for all orientation angles (i.e., to compute the direct Fourier transform, interpolation, matrix product, and inverse Fourier transform) is around 35 sec on a 250-MHz SGI workstation for a 128 × 128 × 10 array. We used the power-of-two Cooley-Tukey FFT algorithm (Elliott and Rao 1982; Cooley and Tukey 1965).

If very precise positioning of the end effector is not required, even regions with density values as low as 10^{-6}th of maximal density value may be important for applications. To emphasize such low densities (which might otherwise be lost when using Fourier methods), and to handle the case of workspaces of continuously actuated manipulators, we augment our approach in the following way. First, for a segment

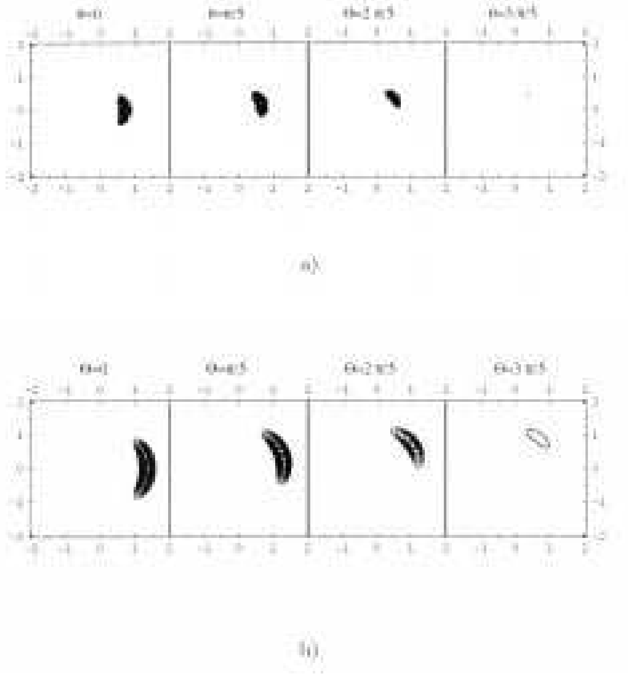


Fig. 5. The workspace of a six-module binary manipulator (computed by brute force) for $\theta = 0 - 3\pi/5$ orientations of the robot (a). The convolved workspace of a 12-module binary manipulator (computed by Fourier transform on the discrete-motion group) for $\theta = 0 - 3\pi/5$ orientations of the robot (b).

[5] Each truss element is either a two-state actuator with stable lengths $l_{\min}$,$l_{\max}$, or a fixed element with length s. Each "module" is composed of three linear actuators and therefore has eight states

[6] "Brute force" means that each of the 8^6 discrete configurations of this arm are calculated, the forward kinematics are computed, and each of the resulting reachable frames are stored in an array corresponding to a discretization of $SE(2)$. The density function is the number of reachable frames divided by the volume of each discretization.

of the manipulator, define a function that is equal to a constant value everywhere where the workspace density of this segment has nonzero value. In this example, we will take the segment to be half of a manipulator consisting of two identical parts. Next we convolve these piecewise-constant functions. The result is a function on the workspace that is much more sensitive to low-density areas for the whole manipulator. The border where such a workspace function crosses zero is depicted in Figure 6 for the orientation angles shown. For a continuously actuated manipulator, this workspace boundary (and not the density information) may be all that is important. If the manipulator were broken into multiple segments, the density function could be redefined after each convolution so as to be constant on the composite segment's workspace and zero otherwise, and the process could be repeated.

As an example to compare the Fourier transform on the discrete-motion group with the continuous case, we computed the convolution of the function

$$f'(r,\phi',\theta) = c_5\,(\frac{1}{\sigma_1\sqrt{2\pi}}\exp(-\frac{(r-x)^2}{2\sigma_1^2}))$$

$$\times(1+c_3\,\cos\theta)^2\,\frac{1}{\sigma_2\sqrt{2\pi}}\exp(-\frac{(\phi'-c_4\,\theta)^2}{2\sigma_2^2}) \qquad (18)$$

with itself, where θ is an orientation angle, and ϕ' is a polar angle of $\mathbf{r}$, $x = 1/2\,(1+\cos\theta)$, $c_1+1/2\,(1-\cos\theta)\,c_2$ (a modified version of this "anzatz" function describes the workspace density of two-dimensional binary manipulators (Kyatkin and Chirikjian, 1999a)). We assume that $-\pi < \theta \leq \pi$, $-\pi < \phi' \leq \pi$, and the density function is assumed to be 2π-periodic. We assume that the orientation angle θ has discrete values $2\pi\, i/N$ $(i = 0, ..., N-1)$.

As an example, in Figure 7a we depict the "slice" $\theta = 0$ for the values of the parameters

$$c_1 = 0.9, \quad c_2 = 0.6, \quad c_3 = 0.8, \quad c_5 = 5.66,$$

$$c_4 = 0.5, \quad \sigma_1 = 0.15, \quad \sigma_2 = 0.46.$$

We consider the C_{10} discrete subgroup of $SO(2)$ $(N = 10)$. For other orientation angles, the function is "rotated" and "shifted" toward the origin, decreasing in magnitude.

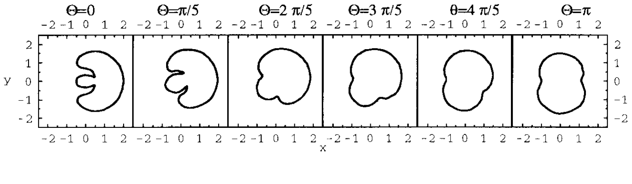


Fig. 6. An approximate boundary of workspace for a 12-module binary manipulator, found by Fourier method.

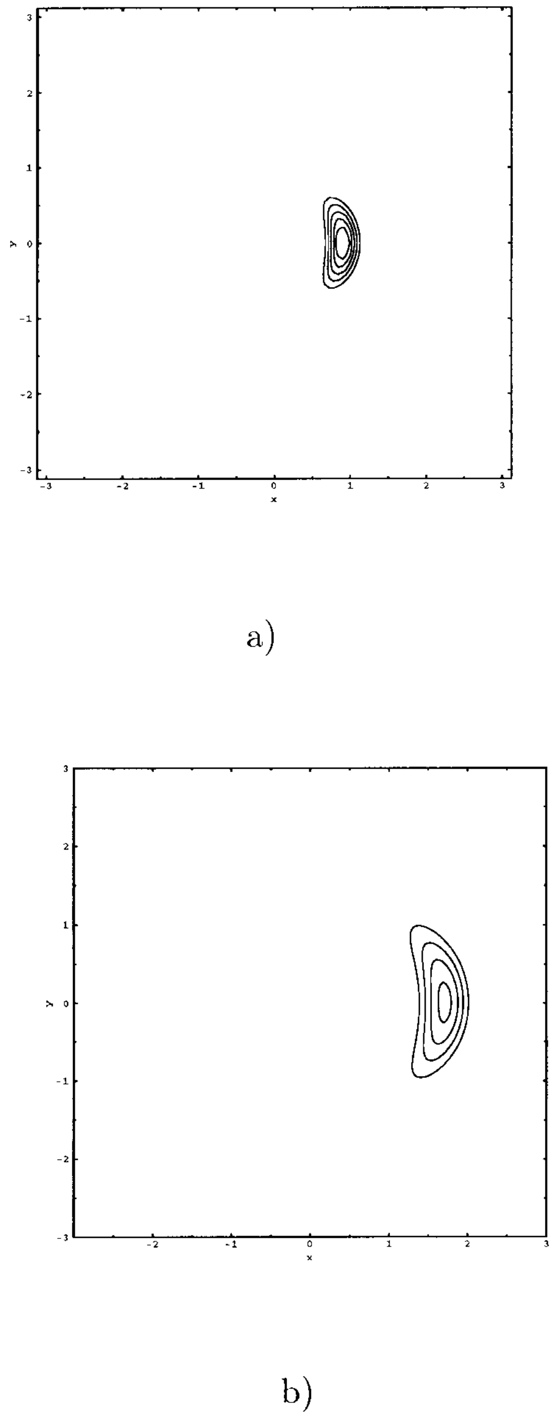


Fig. 7. The contour plot of $f(g)$ for the orientation angle $\theta = 0$ (a); and the contour plot of the convolution $F(g) = (f * f)(g)$ (depicted at $\theta = 0$), which is computed using the FFT (b).

In Figure 7b, we show the convolved function, for $\theta = 0$, computed by the above-described Fourier transform (on a 256×256 grid, $N_v = 20$). Time to compute the convolution for all orientation angles (i.e., computation of the direct Fourier transform, interpolation, matrix product, and inverse Fourier transform) is around 70 sec on a 250-MHz SGI workstation.

We depicted also in Figure 8a the convolution computed by direct integration on the discrete-motion group (time to compute is several hours for each slice). In Figure 8b, the convolution of the continuous function ($N = 200$) computed by the continuous Fourier transform is depicted. The quadratic error of the implemented Fourier transform on the discrete-motion group and the continuous motion group is around 3.7% (the difference between Fig. 8b and Fig. 7b, for all slices). Thus, convolution on the "discrete" motion group approximates accurately convolution on the continuous group when the functions being convolved do not oscillate rapidly. Note that the function $g(\mathbf{x}, A_i)$ computed in the previous section by the Fourier method on the discrete group is identical to the continuous case for the orientations from the C_N group, because the convolution-like integral on the discrete group (eq. (15)) is identical to the integral (eq. (1)) for orientation-independent functions.

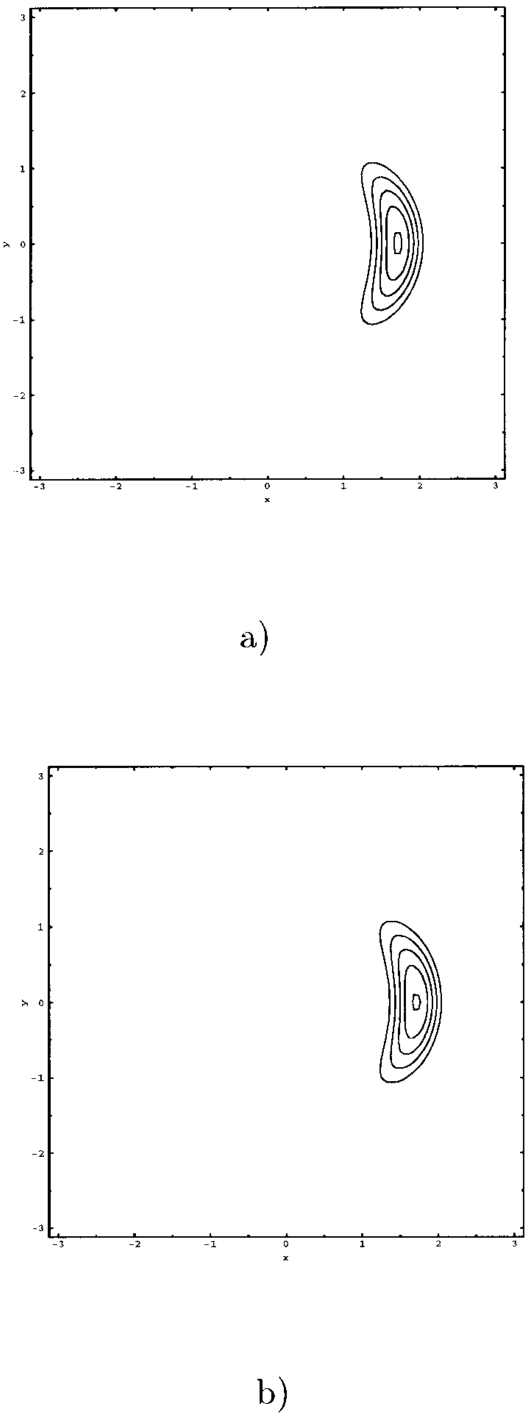
4.3. The Workspace of a Discretely Actuated Revolute Manipulator

In this subsection, we apply the Fourier method to generate the workspace density of the discretely actuated m-link planar revolute manipulator, shown in Figure 2. We assume that each link has one of N orientations (which is an element of C_N) relative to the previous link, and the length of each link L_i has a fixed value. In our example we want to generate the workspace density of a six-link manipulator, assuming that we know the workspace density of each three-link half-manipulator.

We note that the position and orientation of the end effector for such a discretely actuated planar revolute manipulator takes values from the discrete-motion group $G = \mathbb{R}^2 \rhd C_N$. For a small enough number of links, m, each with relatively few states, N, the workspace density of end-effector positions per unit volume of the discrete-motion group may be computed by brute force because N^m configurations can be handled. We assume below that each link may reach 60 possible angles relative to the other link. The gray-level workspace-density values for three-link manipulator parameters,

$$L_1 = 0.2, \quad L_2 = 0.3, \quad L_4 = 0.3 \qquad (19)$$

are depicted in Figure 9 for $\theta = 0$ and $\theta = \pi/2$ orientation angles. The boundary of the workspace (which coincides with the boundary of the workspace for the continuous manipulator) may be also clearly observed.



a)

b)

Fig. 8. The convolution (for $\theta = 0$) of the piecewise-continuous function, computed by direct integration (a); and the convolution of the continuous function computed using the Fourier transform on the continuous-motion group (b).

For a large number of links, the number of brute-force computations increases exponentially. We may use, however, the same method as for the binary vgt manipulator; i.e., partition the manipulator into segments and convolve the workspace densities of these segments. As an example, we generated the workspace density of the six-link manipulator by convolving the workspace density of the three-link manipulator with parameters given by eq. (19) (for the lower part) and the three-link upper half with the parameters

$$L_4 = 0.2, \quad L_5 = 0.3, \quad L_6 = 0.4.$$

We computed the workspace on a 256×256×60 grid using the Fourier method on the discrete-motion group (with a C_{60} rotation subgroup). The whole computation took 15 min on a 250-MHz workstation. The convolved workspace density is depicted in Figure 10 for the orientation angles $\theta = 0$ and $\theta = \pi/2$. The boundary of the workspace also may be observed.

5. Conclusions

This paper develops Fourier analysis on the discrete-motion group and applies the resulting theory to problems in robot kinematics and motion planning. These problems include the fast calculation of mobile-robot configuration-space obstacles, and the workspace-density functions of manipulator arms with finite-state actuators. All of the problems presented

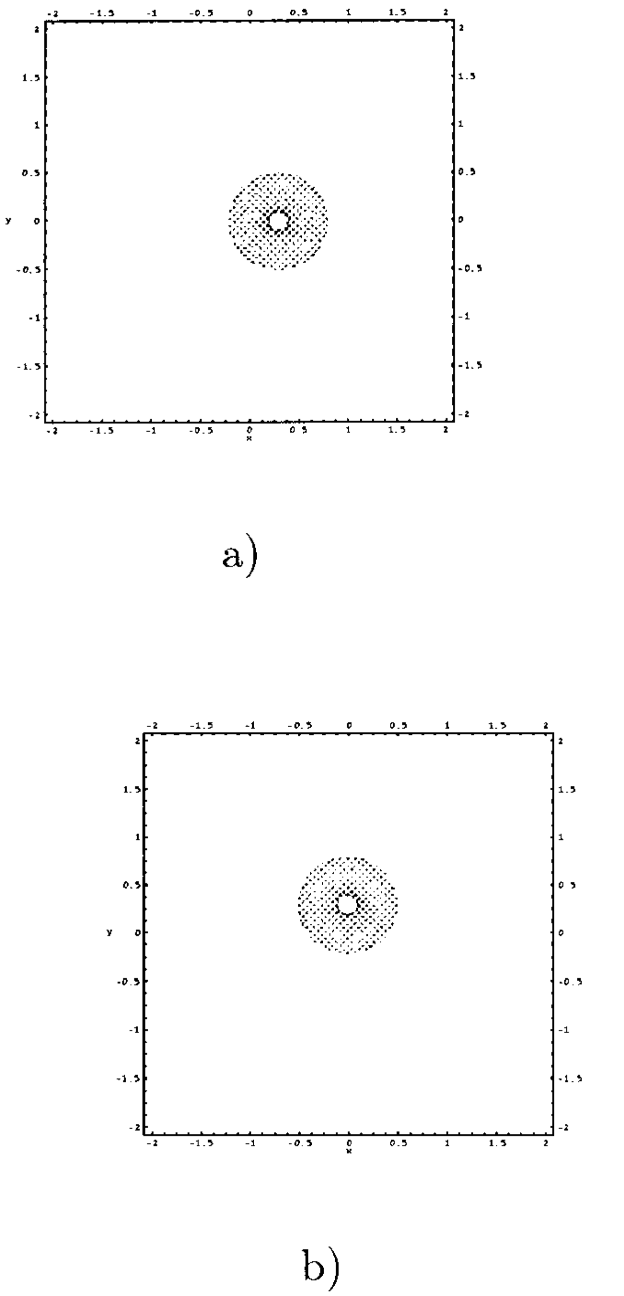


Fig. 9. The gray-level workspace-density values (for $\theta = 0$) of the three-link discretely actuated revolute manipulator, found by brute force (a); and the workspace density for $\theta = \pi/2$ (b).

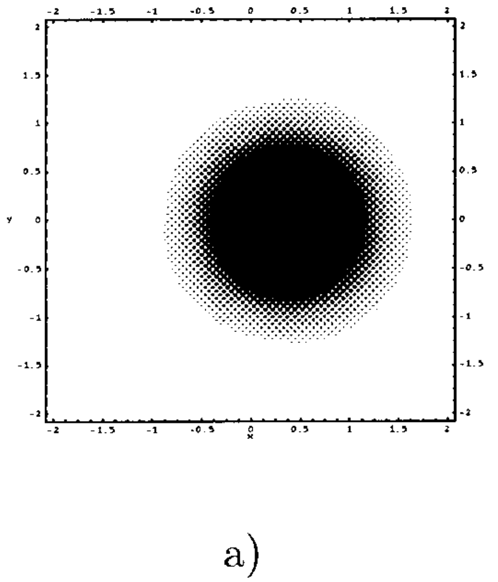


Fig. 10. The workspace density (for $\theta = 0$) of the six-link revolute manipulator, found by convolution on the discrete-motion group using the Fourier method (a); and the convolved workspace density for $\theta = \pi/2$ (b) (see next page).

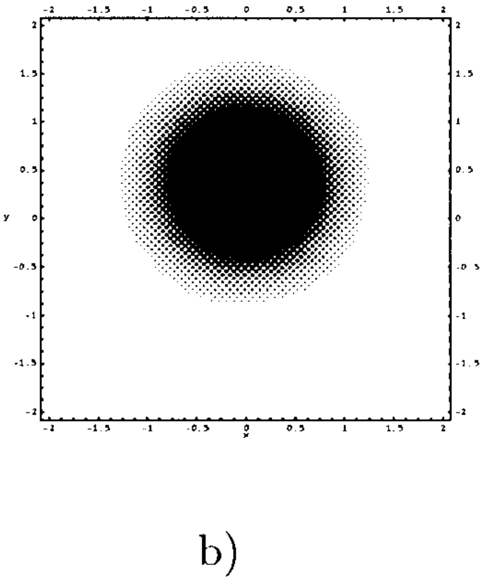


Fig. 10. Continued from previous page.

here can be formulated as Euclidean group convolutions. Using the discrete-motion group as an approximation to the Euclidean group allows us to apply FFT techniques to calculate these generalized convolutions in a very efficient way.

Acknowledgments

This work was supported by the NSF Robotics and Human Augmentation Program with grant IIS 9731720.

References

Basavaraj, U., and Duffy, J. 1993. End-effector motion capabilities of serial manipulators. *Intl. J. Robot. Res.* 12(2):132–145.

Ceccarelli, M., and Vinciguerra, A. 1995. On the workspace of general 4R manipulators. *Intl. J. Robot. Res.* 14(2):152–160.

Chirikjian, G. S. 1994 (San Diego, CA). A binary paradigm for robotic manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 3063–3069.

Chirikjian, G. S. 1995. Kinematic synthesis of mechanisms and robotic manipulators with binary actuators. *ASME J. Mech. Design* 117(Sept.):573–580.

Chirikjian, G. S. 1996. Fredholm integral equations on the Euclidean motion group. *Inverse Problems* 12(Oct.):579–599.

Chirikjian, G. S. 1997. Inverse kinematics of binary manipulators using a continuum model. *J. Intell. and Robot. Sys.* 19:5–22.

Chirikjian, G. S., and Ebert-Uphoff, I. 1998. Numerical convolution on the Euclidean group with applications to workspace generation. *IEEE Trans. Robot. Automat.* 14(1):123–136.

Chirikjian, G. S., and Kyatkin, A. B. Forthcoming. *Engineering Applications of Noncommutative Harmonic Analysis.* Boca Raton, FL: CRC Press. (Scheduled to appear in 2000.)

Cooley, J. W., and Tukey, J. 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* 19:297–301.

Coppersmith, D., and Winograd, S. 1987. Matrix multiplication via arithmetic progressions. *Proc. of the 19th ACM Symp. on the Theory of Computing*. New York: ACM, pp. 1–6.

Ebert-Uphoff, I., and Chirikjian, G. S. 1995. Efficient workspace generation for binary manipulators with many actuators. *J. Robot. Sys.* 12(6):383–400.

Ebert-Uphoff, I., and Chirikjian, G. S., 1996a (April, Minneapolis, MN). Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 139–145.

Ebert-Uphoff, I., and Chirikjian, G. S. 1996b (Aug., Irvine, CA). Discretely actuated manipulator workspace generation by closed-form convolution. *Proc. of the 1996 ASME Design Eng. Tech. Conf. and Comp. in Eng. Conf.* New York: ASME.

Elliott, D. F., and Rao, K. R. 1982. *Fast Transforms: Algorithms, Analyses, Applications.* New York: Academic Press.

Gauthier, J. P., Bornard, G., and Sibermann, M. 1991. Motion and pattern analysis: Harmonic analysis on motion groups and their homogeneous spaces. *IEEE Trans. Sys. Man Cybernet.* 21:159–172.

Guibas, L., Ramshaw, L., and Stolfi, J. 1983. A kinetic framework for computational geometry. *Proc. of the IEEE Symp. on Found. of Comp. Sci.* Los Alamitos, CA: IEEE, pp. 100–111.

Gurarie, D. 1992. *Symmetry and Laplacians. Introduction to Harmonic Analysis, Group Representations, and Applications.* The Netherlands: Elsevier.

Janssen, T. 1973. *Crystallographic Groups.* New York: North-Holland.

Kavraki, L. 1995. Computation of configuration-space obstacles using the fast Fourier transform. *IEEE Trans. Robot. Automat.* 11:408–413.

Koliskor, A. 1986 (Suzdal, former USSR). The l-coordinate approach to the industrial robot's design. *Information Control Problems in Manufacturing Technology: Proc. of the 5th IFAC/IFIP/IMACS/IFORS Conf.*, pp. 225–232. (Preprint).

Kwon, S.-J., Youm, Y., and Chung, K. 1994. General algorithm for automatic generation of the workspace for n-link planar redundant manipulators. *ASME Trans.* 116(Sept):967–969.

Kyatkin, A. B., and Chirikjian, G. S. 1998a (March 5–8, Houston, TX). Fourier methods on groups: Applications in robot kinematics and motion planning. *Proc. of the 3rd Workshop on Algorithmic Found. of Robot.*

Kyatkin, A. B., and Chirikjian, G. S. 1998b. Regularization of a nonlinear convolution equation on the Euclidean group. *Acta Appl. Math.* 53(Aug.):89–123.

Kyatkin, A. B., and Chirikjian, G. S. 1999a. Synthesis of binary manipulators using the Fourier transform on the Euclidean group. *ASME J. Mech. Design* 121:9–14.

Kyatkin, A. B., and Chirikjian, G. S. 1999b. Template matching as a correlation on the motion group. *Computer Vision and Image Understanding* 74(1):22-35.

Lees, D. S., and Chirikjian, G. S. 1996a (April, Minneapolis, MN). An efficient method for computing the forward kinematics of binary manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 1012–1017.

Lees, D. S., and Chirikjian, G. S. 1996b (April, Minneapolis, MN). A combinatorial approach to trajectory planning for binary manipulators. *Proc. of the IEEE Intl. Conf. on Robot. and Automat.* Washington, DC: IEEE, pp. 2749–2754.

Lees, D. S., and Chirikjian, G. S. 1996c (Aug.). An efficient trajectory planning method for binary manipulators. CD-ROM of *Proc. of ASME Mechanisms Conf.*, 96-DETC/MECH-1161 (9 pp.).

Lozano-Perez, T. 1983. Spatial planning: A configuration space approach. *IEEE Trans. Comp.* 32:108–120.

Maslen, D. K. 1993. *Fast transforms and sampling for compact groups*. PhD dissertation, Dept. of Mathematics, Harvard University, Boston, MA.

McCarthy, J. M. 1991. *An Introduction to Theoretical Kinematics.* Cambridge, MA: MIT Press.

Murray, R. M., Li, Z., and Sastry, S. S. 1994. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press.

Newman, W., and Branicky, M. 1991. Real-time configuration space transforms for obstacle avoidance. *Intl. J. Robot. Res.* 10:650–667.

Orihara, A. 1961. Bessel functions and the Euclidean motion group. *Tohoku Math. J.* 13:66–71.

Park, F. C., and Brockett, R. W. 1994. Kinematic dexterity of robotic mechanisms. *Intl. J. Robot. Res.* 13(1):1–15.

Pieper, D. L. 1968 (Oct.). The kinematics of manipulators under computer control. PhD dissertation, Stanford University.

Rockmore, D. N. 1994. Efficient computation of Fourier inversion for finite groups. *J. Assoc. Comp. Machinery* 41:31–66.

Roth, B., Rastegar, J., and Scheinman, V. 1973. On the design of computer controlled manipulators. *First CISM-IFTMM Symp. on Theory and Practice of Robots and Manipulators*. pp. 93–113.

Sugiura, M. 1990. *Unitary representations and harmonic analysis*, 2nd ed. The Netherlands: Elsevier.

Talman, J. 1968. *Special Functions*. Amsterdam: W. A. Benjamin.

Vilenkin, N. J. 1956. Bessel functions and representations of the group of Euclidean motions. *Uspehi Mat. Nauk.* 11:69–112. (In Russian.)