



Workspace density and inverse kinematics for planar serial revolute manipulators

Hui Dong^a, Zhijiang Du^a, Gregory S. Chirikjian^{b,*}

^a State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

^b Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218, USA

ARTICLE INFO

Article history:

Received 13 July 2013

Received in revised form 3 August 2013

Accepted 6 August 2013

Available online 14 September 2013

Keywords:

Manipulator workspaces

Revolute manipulator

Convolution

Inverse kinematics

ABSTRACT

The topic of reachable workspaces of robotic manipulators has received considerable attention over the past half century. One approach to generating workspaces is by sampling joint angles and evaluating the boundary of the resulting set in the space of rigid-body motions. In the case when the manipulator has discrete actuation, such as stepper motors or pneumatic cylinders, not only the boundary of the workspace, but also the density of reachable poses within the workspace is important. Following previous efforts that focused on characterizing this workspace density, we show that this density is particularly efficient to evaluate in the special case of planar serial arms with revolute joints. We then show how the resulting density can be used in inverse kinematics algorithms that are equally applicable for discrete-state and continuous-motion robot arms.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Planar robots are currently used in a number of industrial and medical applications [1]. Moreover, commonly used industrial architectures such as SCARA manipulators like the one shown in Fig. 1 contain planar manipulators as critical components.

The development of algorithms for highly accurate and stable control of planar robotic arms is therefore an important topic. The solution of the inverse kinematics problem is a fundamental part of robot control. Traditionally, three models have been used to solve the inverse kinematics problem. The first is the geometric model, which is well-suited to compute the inverse kinematics of relatively simple manipulators with a small number of links. For example, A. Yu et al. [2] published a geometric approach to the accuracy analysis of class 3-DOF planar parallel robots. The second is the algebraic model, which does not guarantee a “closed-form” solution, but can be efficiently solved by polynomial root finding. D. Manocha and J.F. Canny [3] presented an algorithm and implemented it for efficient inverse kinematics for a general 6R manipulator by extending the polynomial elimination methods of M. Raghavan and B. Roth [4]. The third is the iterative model, the result of which depends on the starting point used. This approach to finding the inverse kinematics solution of robotic manipulators was proposed by J.U. Korein and N.I. Badler [5].

In parallel with intelligent control developments, there are additional novel approaches for solving the inverse kinematics problem. For example, neural-network-based inverse kinematics solution methods for robotic manipulators have been explored recently in [6–12]. For example, B. Karlik and S. Aydin [6] presented a structured artificial neural-network (ANN) to the solution of inverse kinematics problems for a six-degree-of-freedom robot manipulator. Work has been undertaken to find the best ANN configurations for this problem. J.A. Martin et al. [7] proposed a method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers. The method is based on the evolutionary computation paradigm and obtains incrementally better neuro-controllers. R.V. Mayorga and P. Sanongboon [8] presented an ANN approach for fast inverse kinematics computation and effective geometrically bounded singularity prevention of redundant manipulators. E. Oyama et al. [9] proposed a novel expert

* Corresponding author. Tel.: +1 410 516 7127; fax: +1 410 516 7254.

E-mail address: gregc@jhu.edu (G.S. Chirikjian).

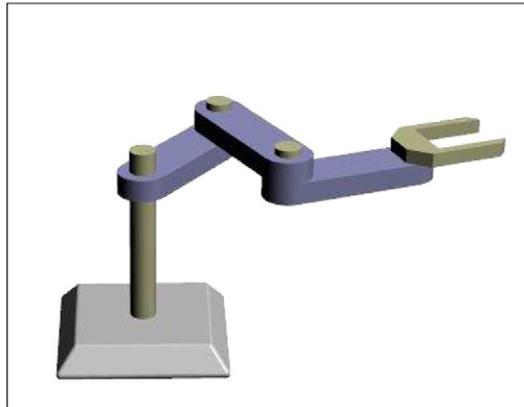


Fig. 1. A SCARA manipulator.

selection by using performance prediction networks which directly calculate the performances of the experts which could reduce the computation time. S.S. Chiddarwar and N.R. Babu [10] presented a fusion approach to determine inverse kinematics solutions of a six degree of freedom serial robot. The effectiveness of the fusion process was shown by comparing the inverse kinematics solutions obtained for an end-effector of an industrial robot moving along a specified path with the solutions obtained from conventional neural network approaches as well as an iterative technique. S. Tejomurtula and S. Kak [11] presented an ANN approach for solving the inverse kinematics problem. The method yielded multiple and precise solutions. It was suitable for real-time applications. S.K. Nanda et al. [12] proposed a novel application of ANNs for the solution of inverse kinematics of robotic manipulators. This method represents the non-linear mapping between Cartesian and joint coordinates using multi layer perceptrons and a functional link artificial neural network.

Genetic algorithm approaches, such as in Refs. [1,13], and [14], have been widely investigated. P. Kalra et al. [1] presented an approach based on an evolutionary genetic algorithm that was used to obtain the solution of the multimodal inverse kinematics problem of industrial robots. A.C. Nearchou [13] used a modified genetic algorithm to search successive robot configurations in the entire free space to specify how the robot should move its end-effector. R. Köker [14] presented a genetic algorithm approach to a neural-network-based inverse kinematics solution for robotic manipulators based on error minimization. In that work, ideas from neural network algorithms and genetic algorithms were fused.

The Jacobian pseudo-inverse approach is a widely used method for solving the inverse kinematics problem. A.A. Maciejewski and C.A. Klein [15] presented the Singular Value Decomposition (SVD) of the Jacobian to compute pseudo-inverse for robotic manipulators. R.G. Roberts and A.A. Maciejewski [16] presented repeatable control strategies that obtain near optimal solutions in the selected workspace.

Researchers have also focused on some other approaches to obtain inverse kinematics solution for robot manipulators. B. Siciliano [17] addressed the inverse kinematics, manipulability analysis, and closed-loop direct kinematics algorithm for the Tricept robot. H. Zhang [18] presented a method to compute inverse kinematics in parallel for robots with a closed form solution. The computational task of inverse kinematics was partitioned with one subtask per joint and all subtasks were computed in parallel. This results in effectiveness and the efficiency of the algorithm for a multiprocessor system. S.R. Lucas et al. [19] compared the merits of many of the methods already presented and described a new approach that led to a fast and numerically well-conditioned algorithm. P. Chiacchio et al. [20] presented new closed-loop schemes for solving the inverse kinematics of constrained redundant manipulators. G.S. Chirikjian and J.W. Burdick [21] presented efficient kinematic modeling techniques for “hyper-redundant” robots. Their methods were based on a backbone curve that captures the hyper-redundant robot’s important macroscopic features. I. Ebert-Uphoff and G.S. Chirikjian [22] introduced algorithms for inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities. They proposed a framework for the discussion of the discretely actuated case and presented an inverse kinematics algorithm. This builds on prior sampling-based approaches to manipulator workspace analysis such as in Refs. [23,24] by observing that sampled workspaces of subchains can be smoothed to result in densities, and these densities can be “added” by convolution to result in the density for the whole manipulator. Y. Wang and G.S. Chirikjian [25] presented workspace generation of hyper-redundant manipulators as a diffusion process. In that work, the evolution of the workspace density function is defined by a diffusion equation, which depends on manipulator length and kinematic properties. A multi-objective optimum design of general 3R manipulators for prescribed workspace limits was proposed by M. Ceccarelli and C. Lanni [26]. In that paper a suitable formulation for the workspace was used for the manipulator design, which was formulated as a multi-objective optimization problem by using the workspace volume and robot dimensions as objective functions, and given workspace limits as constraints.

Each methodology contains certain advantages and disadvantages for solving the inverse kinematics problem of robot manipulators. Our paper derives the “closed-form” workspace density and inverse kinematics for planar serial revolute robot arms.

2. Workspace density

2.1. The kinematics of planar robot arms

A three-link planar revolute robot arm is shown in Fig. 2. Three link lengths specify the geometry of the planar robot, and three angles specify its conformation (or configuration). Let L_1 , L_2 and L_3 denote the lengths of the three links and θ_1 , θ_2 and θ_3 denote the joint angles between the links. The definition for the joint angle θ_i is measured counter clockwise from link $i - 1$ to link i .

The position and orientation of the end effector are presented in the Cartesian coordinate system as

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$\theta = \theta_1 + \theta_2 + \theta_3 \quad (3)$$

and converted to polar coordinates as follows

$$r = \sqrt{x^2 + y^2} \quad (4)$$

$$\phi = \text{Atan2}(y, x). \quad (5)$$

2.2. Gaussian distributions to model positional workspace density

We use a similar approach for workspace generation as that in I. Ebert-Uphoff and G.S. Chirikjian [27]. They used closed-form convolution of real-valued functions on the Special Euclidean Group for workspace generation. A.B. Kyatkin and G.S. Chirikjian also proposed a method based on Fourier transform on the discrete-motion group for computing the configuration and workspaces of manipulators and for design of robots for specified workspaces [28,29].

Roughly speaking, the workspace is the volume within the space of rigid-body motions that the end-effector of the manipulator can reach. Suppose that a robot manipulator has N links, and every joint has M states. The number of points that compose the manipulator workspace is then $K = M^N$. The workspace of the three-link planar robot is shown in Fig. 3, where the vertical direction is the end-effector orientation, θ . Here the length of each link is equal to 1.25. Each joint angle is allowed to

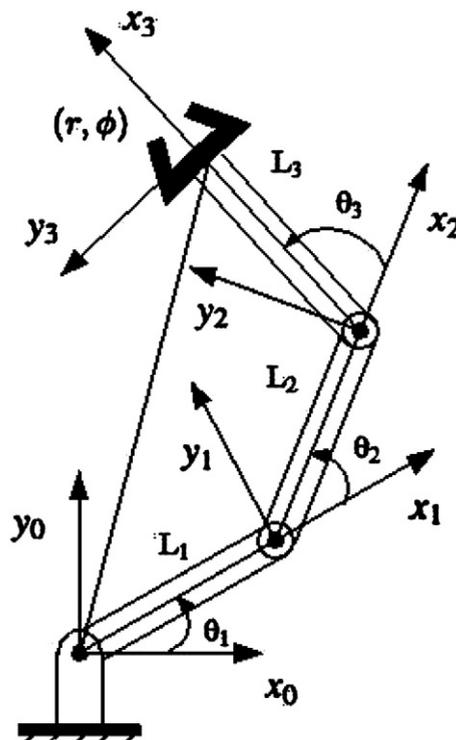


Fig. 2. The kinematic model of a three-link planar robot arm.

change from $-\pi$ to π . There are 40 equally-spaced states in each joint. This manipulator can reach $40^3 = 64,000$ points in the workspace. The color code goes from red to blue, with red denoting high density.

We divide the workspace, W , into small boxes (voxels) of equal size $\Delta x = 0.15, \Delta y = 0.15, \Delta \theta = \frac{\pi}{4}$. The workspace density ρ assigns each box of the workspace W the number of points within the box that are reachable, normalized so as to be a probability density.

$$\rho(\text{box}) = \frac{\# \text{ of reachable points in box}}{(\text{total \# of points}) \cdot (\text{volume of the box})} \tag{6}$$

This density is a probabilistic measure of the positional and orientational (pose) accuracy of the end-effector in a considered area of the workspace. The higher the density in the neighborhood of a point, the more accurately we expect to be able to reach a pose. The positional workspace density of a 3 link planar robot arm (resulting from integrating over the θ direction) is shown in Fig. 4.

As established in [22], when a manipulator is separated into two segments, convolving the densities for each segment will result in the density for the whole manipulator. Generally speaking, if $f_1(g)$ is the density for the lower segment and $f_2(g)$ is the density for the upper one, then the density for the composite will be

$$(f_1 * f_2)(g) = \int_G f_1(h)f_2(h^{-1}g)d(h) \tag{7}$$

where h is a dummy variable of integration in $G = SE(n)$ and $d(h)$ is the Haar measure. This is equally true for the planar ($n = 2$) and spatial ($n = 3$) cases.

In general, if a manipulator has N segments, then

$$f_{1,\dots,N}(g) = (f_1 * f_2 * \dots * f_N)(g), \tag{8}$$

where $*$ is convolution for $G = SE(n)$. Let $g = (R, \mathbf{x}) \in G$. Then $d(g) = dRd\mathbf{x}$ where dR is the Haar measure for $SO(n)$ and $d\mathbf{x}$ is the Lebesgue measure for \mathbb{R}^n .

If we are only interested in the positional density, then we can marginalize $f_{1,\dots,N}(g)$ over $SO(3)$ to result in

$$\rho_{1,\dots,N}(\mathbf{x}) = \int_{SO(n)} f_{1,\dots,N}(R, \mathbf{x}) dR. \tag{9}$$

For $N = 2$ we write out the convolution $f_{1,2}(g) = (f_1 * f_2)(g)$ from Eq. (8) more explicitly as

$$f_{1,2}(R, \mathbf{x}) = \int_{A \in SO(n)} \int_{\mathbf{y} \in \mathbb{R}^n} f_1(A, \mathbf{y}) f_2(A^T R, A^T(\mathbf{x} - \mathbf{y})) dA d\mathbf{y}. \tag{10}$$

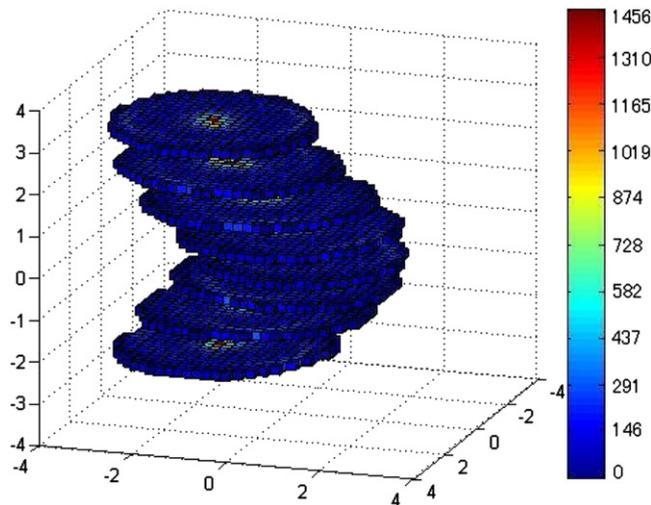


Fig. 3. The workspace of a three-link planar robot with $L_1 = L_2 = L_3 = 1.25$.

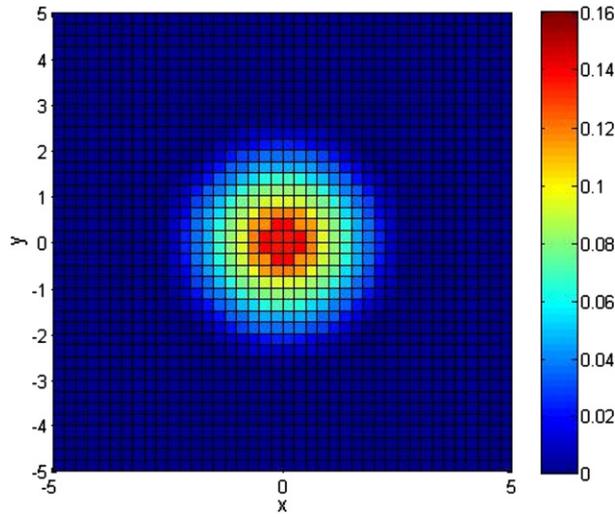


Fig. 4. The workspace density of the three-links planar robot of Fig. 2.

When we substitute Eq. (11) into Eq. (10), the result is

$$\rho_{1,2}(\mathbf{x}) = \int_{A \in SO(n)} \int_{\mathbf{y} \in \mathbb{R}^n} f_1(A, \mathbf{y}) \int_{R \in SO(n)} f_2(A^T R, A^T(\mathbf{x} - \mathbf{y})) dR dA d\mathbf{y}. \tag{11}$$

But

$$\int_{R \in SO(3)} f_2(A^T R, A^T(\mathbf{x} - \mathbf{y})) dR = \rho_2(A^T(\mathbf{x} - \mathbf{y})). \tag{12}$$

In the special case when

$$\rho_i(R\mathbf{x}) = \rho_i(\mathbf{x}), \tag{13}$$

then substituting back into Eq. (12) gives

$$\rho_{1,2}(\mathbf{x}) = (\rho_1 \star \rho_2)(\mathbf{x}) = \int_{\mathbf{y} \in \mathbb{R}^n} \rho_1(\mathbf{y}) \rho_2(\mathbf{x} - \mathbf{y}) d\mathbf{y} \tag{14}$$

where \star is the usual convolution on \mathbb{R}^n . And for N links

$$\rho_{1,\dots,N}(\mathbf{x}) = (\rho_1 \star \rho_2 \star \dots \star \rho_N)(\mathbf{x}). \tag{15}$$

In general, given density functions $\rho_i(\mathbf{x})$ for $i = 1, \dots, N$, the first two moments are defined as

$$\mu_i = \int_{\mathbb{R}^n} \mathbf{x} \rho_i(\mathbf{x}) d\mathbf{x} \tag{16}$$

and

$$\Sigma_i = \int_{\mathbb{R}^n} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T \rho_i(\mathbf{x}) d\mathbf{x}.$$

These are the mean and covariance.

A general property of convolution in \mathbb{R}^n is that

$$\mu_{1,\dots,N} = \sum_{i=1}^N \mu_i \tag{17}$$

$$\Sigma_{1,\dots,N} = \sum_{i=1}^N \Sigma_i. \tag{18}$$

We now examine these moments of pdfs in the special case when the symmetry in Eq. (14) holds, and how these moments behave under convolution. First, note that the mean is

$$\mu_i = \int_{\mathbb{R}^n} \mathbf{x} \rho_i(\mathbf{x}) d\mathbf{x} = \mathbf{0}. \tag{19}$$

And the covariance simplifies as

$$\Sigma_i = \int_{\mathbb{R}^n} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T \rho_i(\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^n} \mathbf{x}\mathbf{x}^T \rho_i(\mathbf{x}) d\mathbf{x} = \sigma_i^2 \mathbb{I}_n \tag{20}$$

where \mathbb{I}_n is the $n \times n$ identity matrix. Then from Eqs. (18) and (19)

$$\mu_{1,\dots,N} = \mathbf{0} \quad \text{and} \quad \Sigma_{1,\dots,N} = \left(\sum_{i=1}^N \sigma_i^2 \right) \mathbb{I}_n. \tag{21}$$

It follows that when all of the links are identical, $\Sigma_{1,\dots,N} = (N\sigma_i^2) \mathbb{I}_n$.

For the N -link planar revolute manipulator (as well as a spatial manipulator with ball-in-socket joints) for which the range of motion is completely unrestricted, the symmetry in Eq. (14) will hold, and the results in Eq. (22) apply. Suppose that for such a manipulator, we sample each joint angle uniformly at M points, resulting in $K = M^N$ samples. The sample covariance for this N -link manipulator is then

$$S_{1,\dots,N}(K) = \frac{1}{K-1} \sum_{k=1}^K \mathbf{x}_k \mathbf{x}_k^T. \tag{22}$$

If M is large enough, then this sample covariance, will become the true covariance:

$$\lim_{M \rightarrow \infty} S_{1,\dots,N}(K) = \Sigma_{1,\dots,N}.$$

If we compute (or obtain a sample estimate) of σ_i for each link in a manipulator, using the above equations we can obtain the covariance for the whole manipulator, $\Sigma_{1,\dots,N}$. Since, the central limit theorem states that iterated convolutions results in a

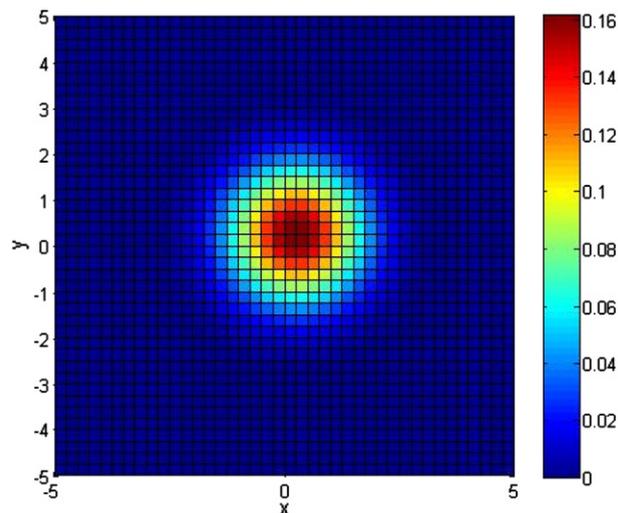


Fig. 5. The workspace density modeled as a Gaussian distribution.

Gaussian density function,

$$\rho(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \quad (23)$$

we can use this to describe the positional workspace of a manipulator for the case when Eq. (16) applies.

In the planar ($n = 2$) case, we get

$$S_{1,\dots,N}(K) \approx \boldsymbol{\Sigma}_{1,\dots,N} = \begin{pmatrix} \sigma_{1,\dots,N}^2 & 0 \\ 0 & \sigma_{1,\dots,N}^2 \end{pmatrix} \quad (24)$$

$$\sigma_{1,\dots,N}^2 \approx \frac{1}{2} \text{tr}(S_{1,\dots,N}). \quad (25)$$

If all of the links are the same, then $\sigma_{1,\dots,N}^2 = N\sigma_1^2$. Substituting Eqs. (25) and (26) into Eq. (24) in this case, the resulting density is

$$\rho_{1,\dots,N}(\mathbf{x}) = \frac{1}{2\pi\sigma_{1,\dots,N}^2} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma_{1,\dots,N}^2}\right) \quad (26)$$

$$= \frac{1}{2\pi N\sigma_1^2} \exp\left(-\frac{\|\mathbf{x}\|^2}{2N\sigma_1^2}\right). \quad (27)$$

The positional workspace corresponding to the sampled version in Fig. 4, when modeled as a Gaussian distribution, is shown in Fig. 5. In both figures, $N = 3$.

As can be seen from these figures, the Gaussian captures the total positional density quite well. But for many practical applications, it is desirably to know the full pose density. The generation of such densities is the topic of the next section.

2.3. Using the motion-group Fourier transform to compute workspace density

In prior works mentioned earlier, the Fourier transform for the group of rigid-body motions of the plane, $SE(2)$, has been used to compute workspace densities. The novelty observed in the present work is that the workspace densities for revolute manipulators can be written in a form that provides special structure, giving these densities closed-form expressions.

2.3.1. Joint angles without stops

From Fig. 1, in the case of a single link we have $\phi = \theta$ and $r = L_1 = L$. The workspace density function of one link with one freely moves revolute joint at its base is expressed as

$$f_1(\mathbf{g}; L) = \frac{1}{2\pi} \frac{\delta(r-L)}{r} \delta(\theta-\phi) \quad (28)$$

where $-\pi \leq \theta, \phi \leq \pi$. The δ function, or the Dirac delta function, is a generalized function on the real number line that is zero everywhere except at zero, with an integral of one over the entire real line. As a result, the integral of the function $f_1(\mathbf{g}; L)$ over x , y , and θ with respect to the measure $d(\mathbf{g}) = dx dy d\theta$ (or equivalently, over r, ϕ, θ with respect to the measure $d(\mathbf{g}) = r dr d\phi d\theta$) has a value of unity. Here $g(r, \phi, \theta)$ is the homogeneous transformation matrix with translations parameterized expressed in polar coordinates,

$$g(r, \phi, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta & r \cos\phi \\ \sin\theta & \cos\theta & r \sin\phi \\ 0 & 0 & 1 \end{pmatrix}. \quad (29)$$

The Fourier transform of a workspace density function, $f(\mathbf{g})$, is an infinite-dimensional matrix defined as

$$F(f) = \hat{f}(p) = \int_G f(\mathbf{g}) U(\mathbf{g}^{-1}, p) d(\mathbf{g}) \quad (30)$$

where $U(g^{-1}, p)$ is a unitary representation of $G = SE(2)$ and $d(g)$ is the natural bi-invariant integration measure for $SE(2)$, and “ p ” is a frequency-like parameter. In group theory, the set of all “ p ” values is called the “unitary dual” of G , and is denoted as \widehat{G} and when $G = SE(2)$, $\widehat{G} = \mathbb{R}_{>0}$. The matrix elements of this representation are expressed as

$$u_{mn}(g^{-1}, p) = i^{n-m} e^{im\theta + (n-m)\phi} J_{m-n}(pr). \tag{31}$$

where $i = \sqrt{-1}$ and $J_{m-n}(x)$ is the $(m - n)$ th Bessel function, which we evaluate at $x = p \cdot r$.

The matrix elements of the Fourier transform of this function are

$$\widehat{f}_{mn}(p; L) = \int_G f_1(g) u_{mn}(g^{-1}, p) r dr d\phi d\theta = i^{n-m} J_{m-n}(pL) \int_{\theta=0}^{2\pi} \widetilde{f}_1(\theta) e^{in\theta} d\theta \tag{32}$$

where $\widetilde{f}_1(\theta) = \frac{1}{2\pi}$, and so

$$\int_0^{2\pi} \widetilde{f}_1(\theta) e^{in\theta} d\theta = \delta_{0,n}. \tag{33}$$

The Kronecker delta function $\delta_{0,n}$ is defined for integer variables m and n such that

$$\delta_{m,n} = \begin{cases} 0 & \text{if } n \neq m \\ 1 & \text{if } n = m \end{cases}. \tag{34}$$

Due to the structure of the density function for a single-link manipulator, the matrix elements of the Fourier transform \widehat{f}_{mn} simplify to

$$\begin{aligned} \widehat{f}_{mn}(p; L) &= i^{n-m} \delta_{0,n} J_{m-n}(pL) \\ &= i^{n-m} \delta_{0,n} J_m(pL). \end{aligned} \tag{35}$$

The $m - k$ th element of the squared Fourier transform matrix $\widehat{f}^2(p; L)$ is

$$\begin{aligned} (\widehat{f}^2(p; L))_{mk} &= \sum_{n=-\infty}^{\infty} \widehat{f}_{mn}(p; L) \widehat{f}_{nk}(p; L) \\ &= \sum_{n=-\infty}^{\infty} (i^{n-m} \delta_{0,n} J_m(pL)) \cdot (i^{k-n} \delta_{0,k} J_n(pL)) \\ &= i^{k-m} J_m(pL) J_0(pL) \delta_{0,k} = J_0(pL) \widehat{f}_{mk}(p; L). \end{aligned} \tag{36}$$

From the above derivation, it is not difficult to see that for the N -link case, the Fourier transform matrix has elements

$$(\widehat{f}^N(p; L))_{mn} = (J_0(pL))^{N-1} \widehat{f}_{mn}(p; L). \tag{37}$$

In general, the inverse Fourier transform (IFT) corresponding to Eq. (31) is

$$F^{-1}(\widehat{f}) = f(g) = C \cdot \int_0^{\infty} \text{trace}(\widehat{f}(p; L) U(g, p)) p dp \tag{38}$$

where $C = \frac{1}{(2\pi)^2}$.

The workspace density for an N -link manipulator with equal link lengths is denoted as $f_N(g; L)$. When link lengths are different, then we write $f_N(g; L_1, \dots, L_N)$ for an N -link manipulator. In this way $f_N(g; L)$ is shorthand for $f_N(g; L, L, \dots, L)$. The fact that this Fourier transform matrix with elements $(\widehat{f}^N(p; L))_{mn}$ is used to reconstruct $f_N(g; L)$ with the inversion formula is seen as follows:

$$f_N(g; L) = (f * f * f \cdots * f)(g; L) = C \cdot \int_0^{\infty} (J_0(pL))^{N-1} \text{tr}(\widehat{f}(g; L) U(g, p)) p dp \tag{39}$$

where:

$$\text{tr}(\widehat{f}U) = \sum_{m,n=-\infty}^{\infty} \widehat{f}_{nm} u_{mn} = \sum_{m,n=-\infty}^{\infty} (i^{m-n} \delta_{0,m} J_n(pL)) \cdot u_{mn} = \sum_{n=-\infty}^{\infty} i^{-n} J_n(pL) u_{0,n}(g, p). \tag{40}$$

In the above formula, the length of each link is L . Based on Eq. (36) the $m - k$ th element of the squared Fourier transform matrix $\widehat{f}^2(p; L_1, L_2)$ for different link lengths is

$$\begin{aligned} \left(\widehat{f}^2(p; L_1, L_2)\right)_{mk} &= \sum_{n=-\infty}^{\infty} \widehat{f}_{mn}(p; L_1) \widehat{f}_{nk}(p; L_2) = \sum_{n=-\infty}^{\infty} \left(i^{n-m} \delta_{0,n} J_m(pL_1)\right) \cdot \left(i^{k-n} \delta_{0,k} J_n(pL_2)\right) = i^{k-m} J_m(pL_1) J_0(pL_2) \delta_{0,k} \\ &= J_0(pL_2) \widehat{f}_{mk}(p; L_1). \end{aligned} \tag{41}$$

From the above derivation, it is not difficult to see that for the N -link case, the Fourier transform matrix has elements

$$\left(\widehat{f}^N(p; L)\right)_{mn} = \prod_{i=2}^N J_0(pL_i) \widehat{f}_{mn}(p; L_1). \tag{42}$$

Therefore, the “closed-form” workspace density when the link lengths are all different is:

$$f_N(g; L_1, \dots, L_N) = C \cdot \int_0^{\infty} \prod_{i=2}^N J_0(pL_i) \text{tr}\left(\widehat{f}(g; L_1) U(g, p)\right) p dp. \tag{43}$$

2.3.2. Modeling joint limits

Suppose that instead of swinging all the way around, the joint angles are limited to the range $-\theta_0$ to θ_0 . For a single link, it is still the case that $\theta = \phi$ and $r = L$. The workspace density function of a revolute manipulator for one link is expressed as

$$f(g(r, \phi, \theta)) = \widetilde{f}_1(\theta) \frac{\delta(r-L)}{r} \delta(\theta-\phi) \tag{44}$$

where:

$$\widetilde{f}_1(\theta) = \begin{cases} \frac{1}{2\theta_0} & \text{if } -\theta_0 \leq \theta \leq \theta_0 \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

Based on Eq. (33), the matrix elements of the Fourier transform of this function are

$$\widehat{f}_{mn}(p; L) = i^{m-n} \frac{\sin(n\theta_0)}{n\theta_0} J_{m-n}(pL). \tag{46}$$

The resulting Fourier matrix can be decomposed as

$$\widehat{f}(p; L) = QVQ^{-1}. \tag{47}$$

The Fourier transform matrix is computed based on Eq. (48). Here we truncate at $-M \leq m, n \leq M$, and the resulting Fourier matrix is $(2M + 1) \times (2M + 1)$. The SE(2) Fourier transform of the density function for the N -link revolute manipulator then can be computed as

$$\left[\widehat{f}(p; L)\right]^N = Q V^N Q^{-1}. \tag{48}$$

The inverse Fourier transform for the N -link revolute manipulator density can be written in terms of elements as

$$f_N(g; L) = \sum_{n,m \in \mathbb{Z}} \int_0^{\infty} \left[\widehat{f}^N(p; L)\right]_{mn} u_{nm}(g, p) p dp. \tag{49}$$

In the numerical evaluation of this reconstruction formula we truncate the range of the integral to $[0, P]$ and the sum to $-M \leq n, m \leq M$ where $P = M = 10$, and use an integration step for p of $\Delta p = 0.01$. Using the cutoff $M = 10$ specifies the size of Fourier transform matrix as 21×21 .

2.3.3. Numerical computation of workspace density

Let each link length be $L = 1.25$ and the number of links be $N = 3$. For the sake of numerical evaluation, sample each joint uniformly on its range with 1000 states from $-\pi$ to π . For the purpose of display, slice θ into four parts so that the thickness of each parts is $\frac{\pi}{4}$. Fig. 6(b) (e) (h) and (k) show the discrete workspace density and the corresponding density of this manipulator with three links generated from the Fourier method based on Eq. (40). Fig. 6(a) (b) and (c) compare the workspace density of $\theta \in [-\pi, -\frac{\pi}{2}]$. In Fig. 6(d) (e) and (f), $\theta \in [-\frac{\pi}{2}, 0]$. In Fig. 6(g) (h) and (i), $\theta \in [0, \frac{\pi}{2}]$. In Fig. 6(j) (k) and (l), $\theta \in [\frac{\pi}{2}, \pi]$.

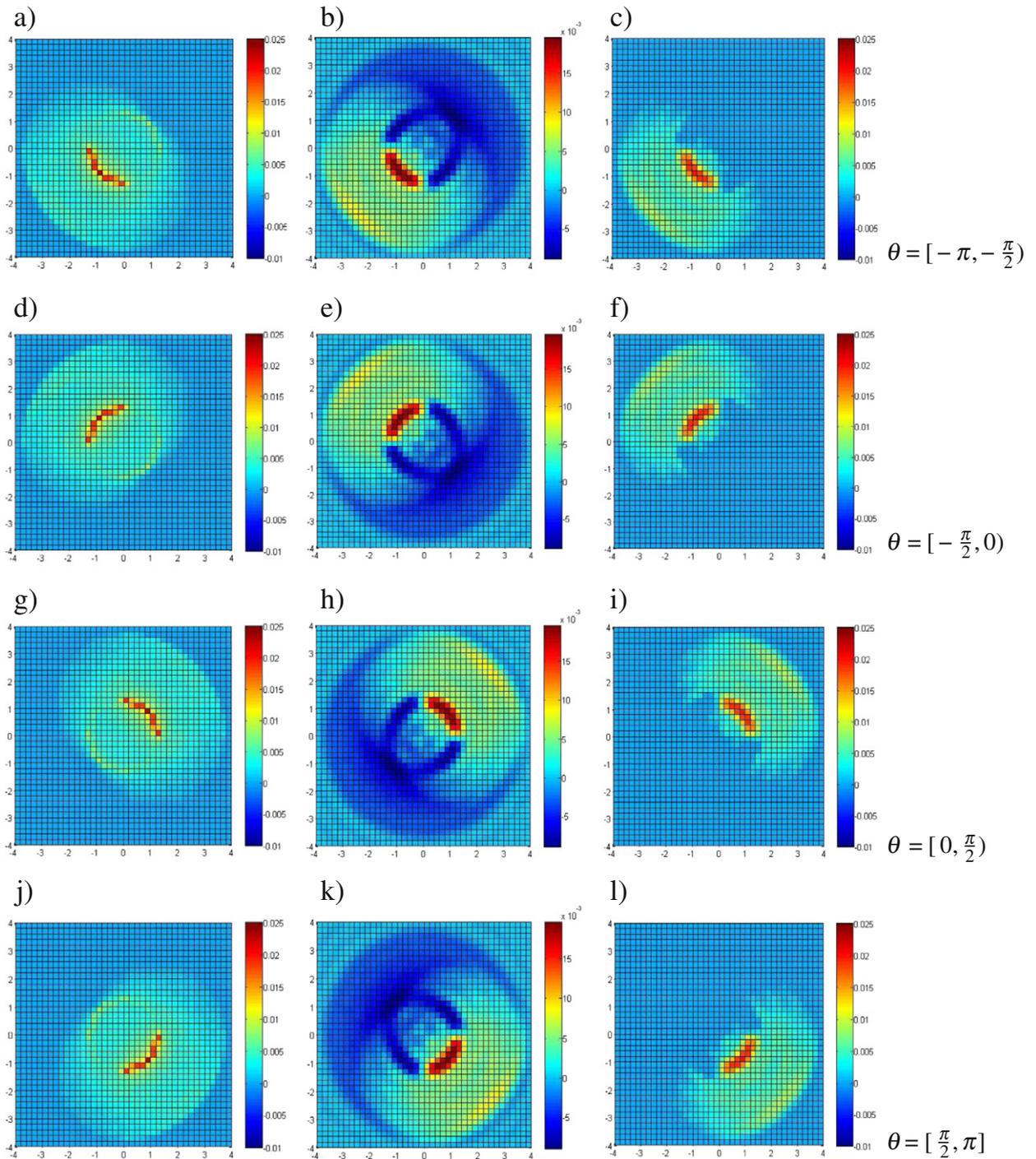


Fig. 6. Comparison of the sample-based workspace density with the corresponding density of this manipulator with three links generated from the Fourier transform.

Fig. 6 compares the discrete workspace density with the corresponding density of this manipulator with three modules generated from the $SE(2)$ Fourier transform. As can be seen, the maximal values of density appear in the same areas. In the left column (panels (a), (d), (g), (j)) is the density obtained from sampling. In the middle column is the result of the Fourier method (panels (b), (e), (h), (k)). In the right column (panels (c), (f), (i), (l)) is the Fourier method with negative values set to zero.

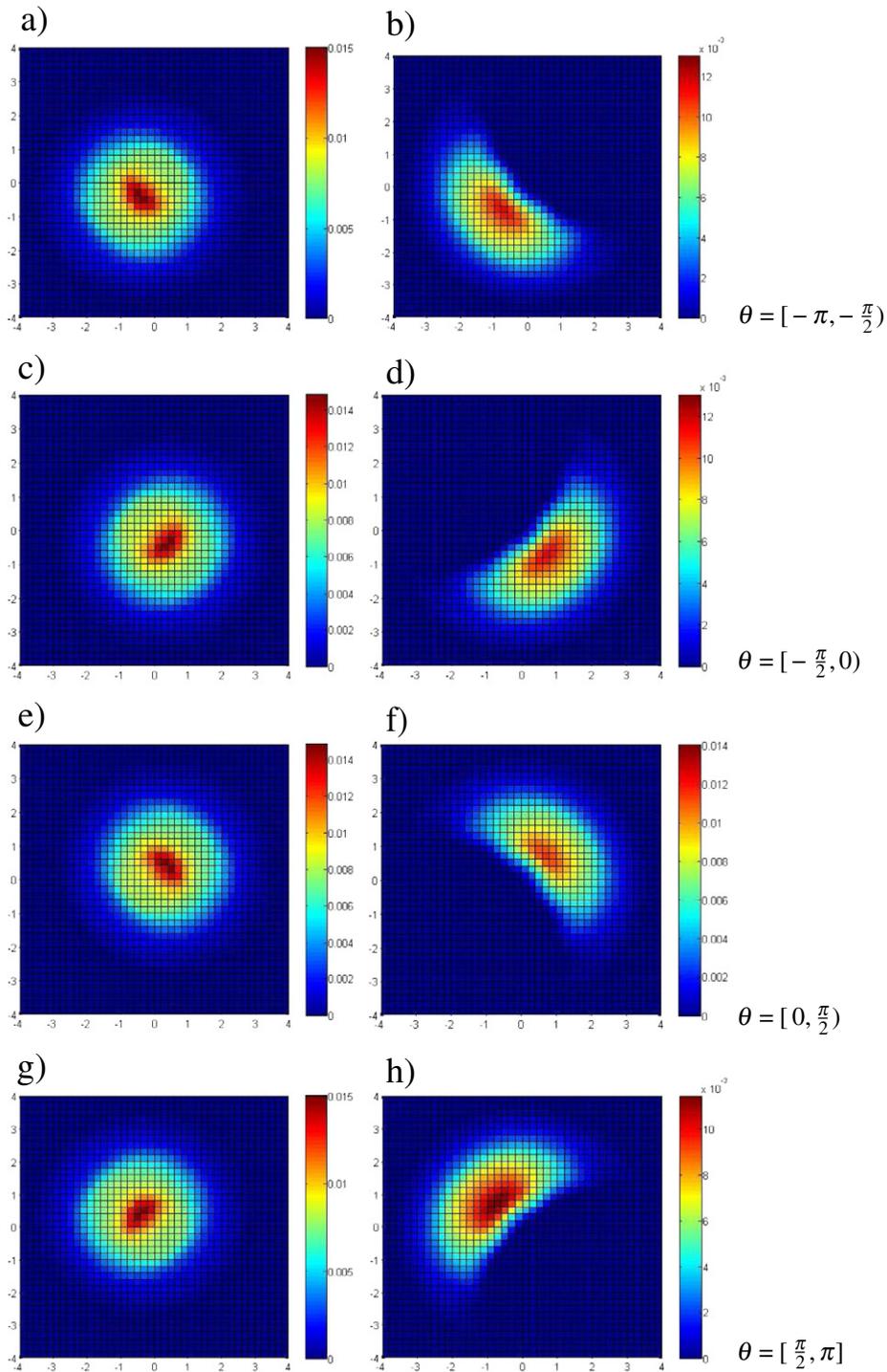


Fig. 7. Comparison of the sample-based workspace density with the corresponding density of this manipulator with five different link lengths generated from the Fourier transform.

In order to illustrate the case of different link lengths, we take $L_n = Le^{-an}$ where $L = 1$ and $a = 0.1$. In this way, the link lengths gradually decrease. Based on Eq. (44), the simulation of five links is shown in Fig. 7. In the left column (panels (a), (c), (e), (g)) is the density obtained from sampling. In the right column (subfigures (b), (d), (f), (h)) is the result of the Fourier method.

As another case, consider when the joint angles are limited to $\theta \in [-\theta_0, \theta_0]$. The Fourier-based approach to computing workspace density of revolute manipulators is given by Eq. (50). We set the length of each link as $L = 1.25$, the number of links to

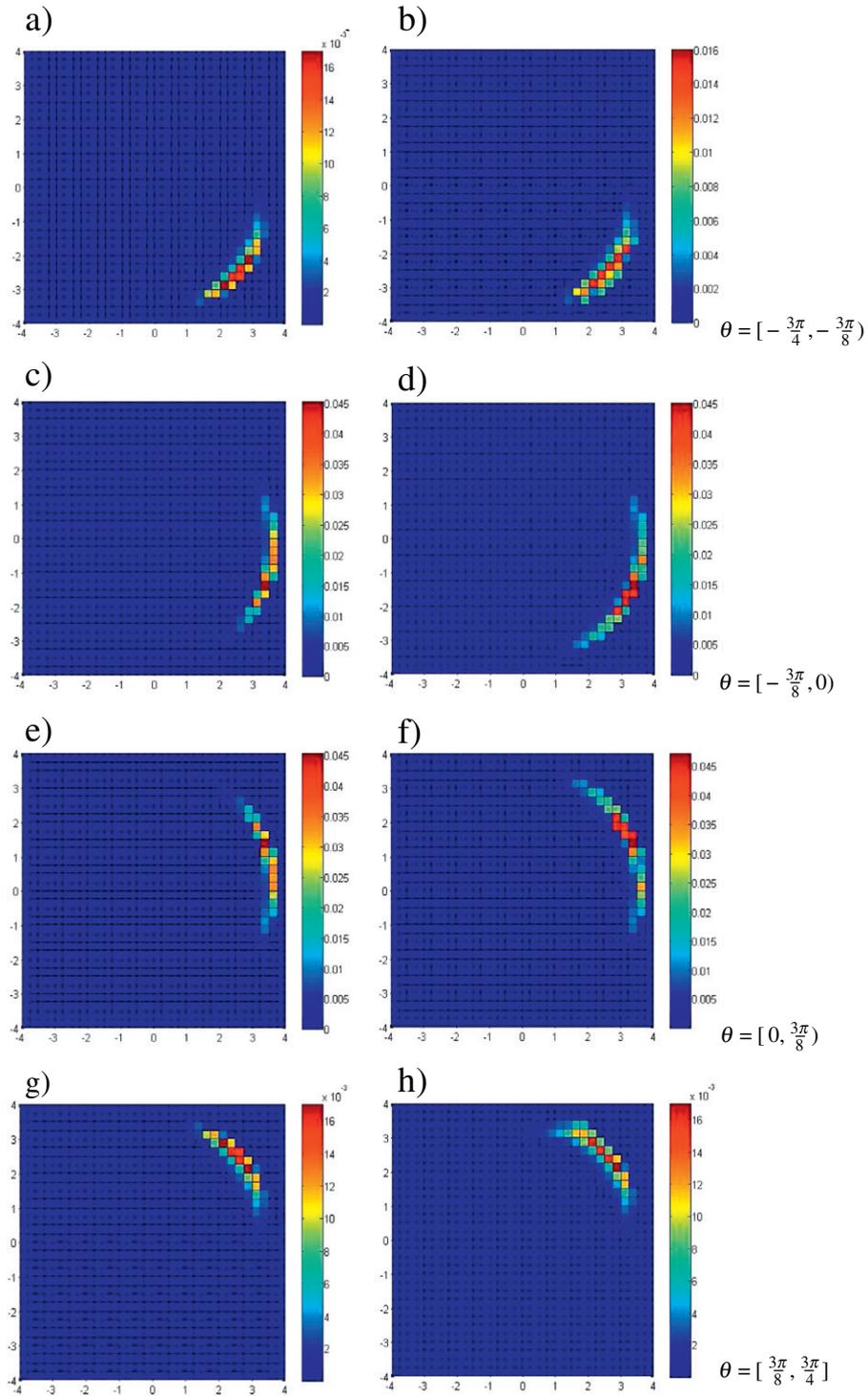


Fig. 8. Comparison of the workspace density generated by sampling and Fourier methods for a manipulator with three links.

$N = 3$, and we set $\theta_0 = \frac{\pi}{4}$. The resulting workspace densities of the manipulator are compared in Fig. 8. Slice the θ into four parts. In Fig. 8(a) and (b) $\theta \in [-\frac{3\pi}{4}, -\frac{3\pi}{8}]$. In Fig. 8(c) and (d) $\theta \in [-\frac{3\pi}{8}, 0]$. In Fig. 8(e) and (f) $\theta \in [0, \frac{3\pi}{8}]$. In Fig. 8(g) and (h) $\theta \in [\frac{3\pi}{8}, \frac{3\pi}{4}]$. In the left column (panels (a), (c), (e), (g)) is the density obtained from sampling. In the right column (panels (b), (d), (f), (h)) is the result of the Fourier method.

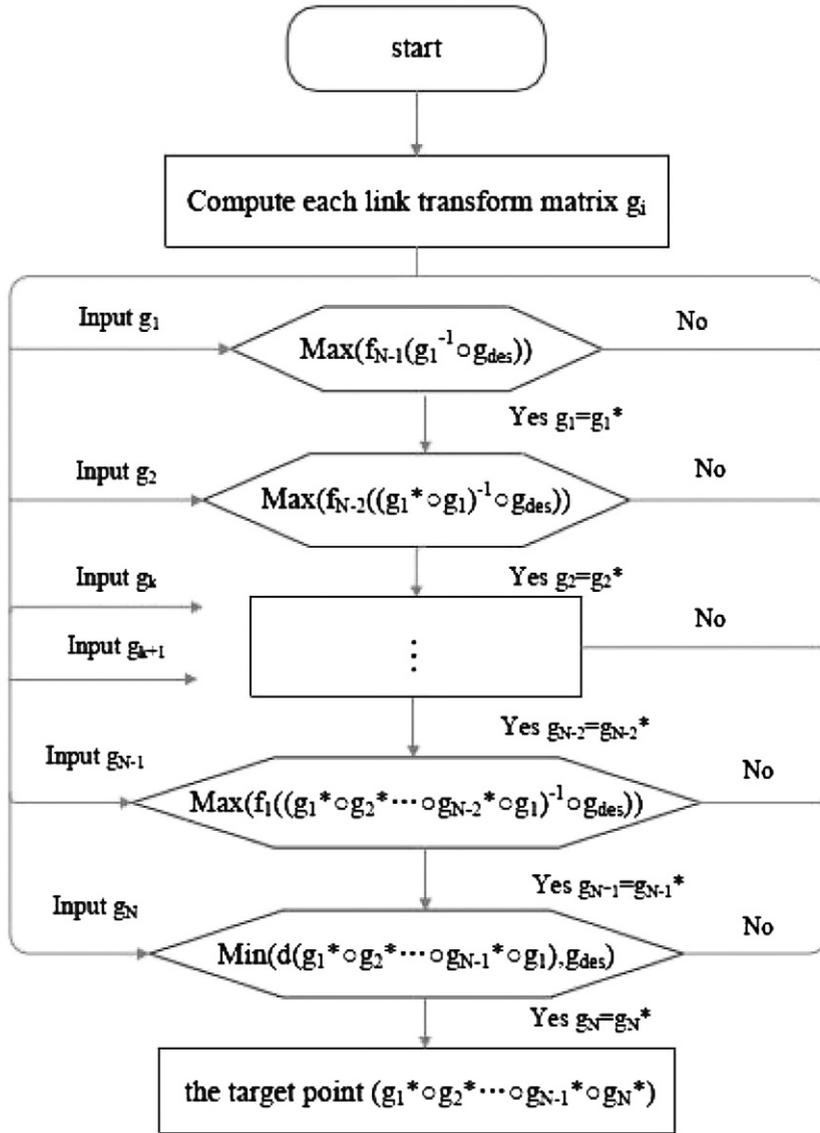


Fig. 9. Flowchart for the inverse kinematics method.

3. Inverse kinematics using workspace density

To illustrate the usefulness of our Fourier-based approach to workspace density generation, we solve the inverse kinematics problem by workspace density generated using this approach. This method is similar to the J. Suthakorn and G. S. Chirikjian [30] approach which presented an inverse kinematics algorithm for binary manipulators with many actuators. The criterion of this method is to select joint angles so as to obtain the maximum workspace density near the target point to fix the configuration of the manipulator.

Consider a discretely actuated serial revolute manipulator with N links. Let g_k denote the homogenous transformation from the base of the k th link to its own distal end, where $k \in \{1, 2, \dots, N\}$. The homogenous transformation from the base of the manipulator to the distal end of k th link is denoted $g^{(k)}$.

$$g^{(k)} = g_1 \circ g_2 \circ \dots \circ g_k. \quad (50)$$

The homogenous transformation from the base of k th link to the distal end of the manipulator is

$$(g^{(k)})^{-1} \circ g^{(N)} = g_{k+1} \circ g_{k+2} \circ \dots \circ g_N. \quad (51)$$

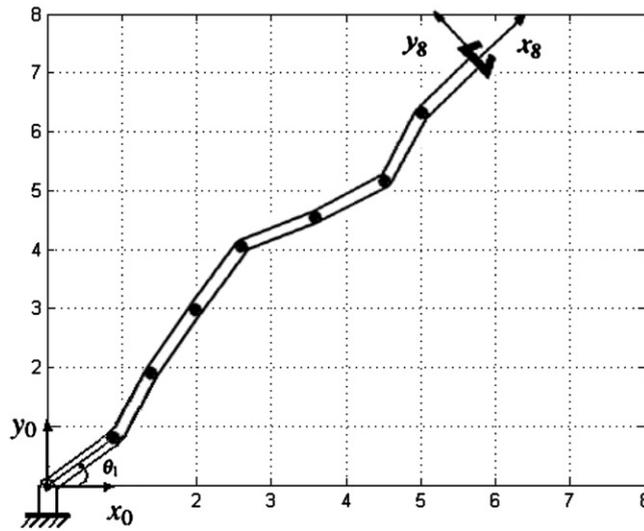


Fig. 10. A simulation result of the inverse kinematics method.

The inverse kinematics method based on workspace density generated using the Fourier method is shown in Fig. 9. In Fig. 9, g_{des} denotes the target pose. All of the possible states of one joint are computed. The transformations between them are denoted by g_k , where $k \in \{1, 2, \dots, N\}$. For the k th link, we find g_k which makes this link maximize the workspace density $f_{N-k}((g_1^* \circ g_2^* \circ \dots \circ g_k)^{-1} \circ g_{des})$. Then we fix the transformation of the k th link to g_k^* . Then we proceed up the manipulator one link at a time. Among all possible states, we search the g_{k+1} that can achieve the highest density $f_{N-k-1}((g_1^* \circ g_2^* \circ \dots \circ g_k^* \circ g_{k+1})^{-1} \circ g_{des})$. If g_{k+1} is such a state, we configure the $k+1$ st module to g_{k+1}^* . This procedure is performed by sequentially maximizing density for the first to $(N-1)$ st link. At the last step, we compute the minimum distance between the pose of N th link and the target pose, and fix g_N^* .

4. Numerical simulations for inverse kinematics

The manipulator with 8 links is used to illustrate the inverse kinematics approach. The length of each link is $L = 1.25$, and the joint angle is unlimited $\theta \in [-\pi, \pi]$. We give the pose $(x_{des}, y_{des}, \theta_{des}) = (6, 7.5, \frac{\pi}{3})$ defining g_{des} . The inverse kinematics algorithm in Section 3 is now demonstrated with the workspace density from Eq. (40). The simulation result is shown in Fig. 10. The segmented lines display the corresponding configurations of the manipulator, where each segment stands for a link.

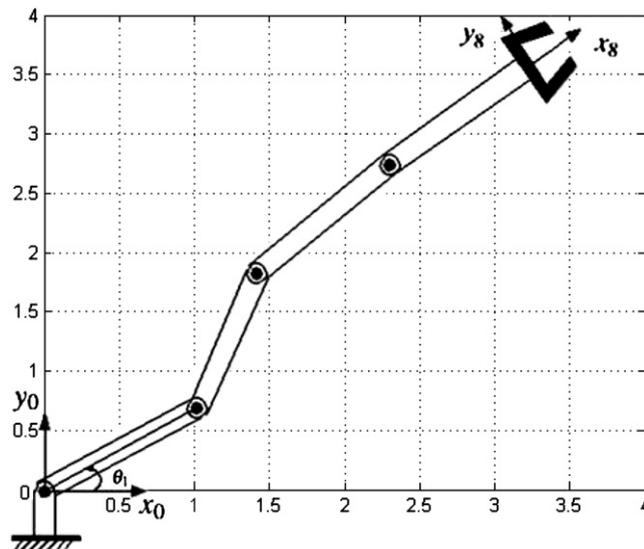


Fig. 11. Another simulation result of the inverse kinematics method.

For the limited joint angles, the manipulator with four links is used to simulate the inverse kinematics approach. The inverse kinematics algorithm uses the workspace density generated by Eq. (50). The length of each link is again $L = 1.25$, and the joint angle is limited $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. We give the target pose as $(x_{des}, y_{des}, \theta_{des}) = (3, 3.5, \frac{\pi}{8})$. The simulation result is shown in Fig. 11.

From all of these tests, we see that the inverse kinematics algorithm using the workspace density generated by the Fourier method provides an accurate solution to reach the target.

5. Conclusion

By using a combination of the concept of the Fourier transform for the group of rigid-body motions of the plane, the resulting convolution theorem, and the particular form of the workspace density of a single link in a planar revolute manipulator, we show that the workspace density for planar serial revolute manipulators can be computed efficiently. This density is then used to solve the inverse kinematics problem for these manipulators. The significance of this approach is that whereas methods based on Jacobian pseudo-inverses assume continuous motion and the differentiability of forward kinematics, the approach taken here selects a solution from among a very large discrete set using workspace density as an evaluation criterion. Challenging mathematical issues remain in the adaptation of this method to the three-dimensional case, though the conceptual framework is the same.

Acknowledgements

The authors would like to thank China Scholarship Council for grant No. 201206120127 that supported Ms. Hui Dong's visit at JHU, and the US National Science Foundation for Grant IIS-1162095 that supported G. Chirikjian.

References

- [1] P. Kalra, P.B. Mahapatra, D.K. Aggarwal, An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots, *Mech. Mach. Theory* 41 (2006) 1213–1229.
- [2] A. Yu, I.A. Bonev, P. Zsombor-Murray, Geometric approach to the accuracy analysis of a class of 3-DOF planar parallel robots, *Mech. Mach. Theory* 43 (2008) 364–375.
- [3] D. Manocha, J.F. Canny, Efficient inverse kinematics for general 6R manipulators, *IEEE Trans. Robot. Autom.* 10 (5) (1994) 648–657.
- [4] M. Raghavan, B. Roth, Inverse kinematics of the general 6R manipulator and related linkages, *J. Mech. Des.* 115 (3) (Sept 01, 1993) 502–508.
- [5] J.U. Korein, N.I. Badler, Techniques for generating the goal-directed motion of articulated structures, *IEEE Comput. Graph. Appl.* 2 (9) (1982) 71–81.
- [6] B. Karlik, S. Aydin, An improved approach to the solution of inverse kinematics problems for robot manipulators, *Eng. Appl. Artif. Intell.* 13 (2000) 159–164.
- [7] J.A. Martin, J.D. Lope, M. Santos, A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers, *Neurocomputing* 72 (2009) 2806–2814.
- [8] R.V. Mayorga, P. Sanongboon, Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: an artificial neural network approach, *Robot. Auton. Syst.* 53 (2005) 164–176.
- [9] E. Oyama, N.Y. Chong, A. Agah, Inverse kinematics learning by modular architecture neural networks with performance prediction networks, *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, 5, 2001, pp. 21–26.
- [10] S.S. Chiddarwar, N.R. Babu, Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach, *Eng. Appl. Artif. Intell.* 23 (7) (2010) 1083–1092.
- [11] S. Tejomurtula, S. Kak, Inverse kinematics in robotics using neural networks, *Inf. Sci.* 116 (1999) 147–164.
- [12] S.K. Nanda, S. Panda, P.R.S. Subudhi, R.K. Das, A novel application of artificial neural network for the solution of inverse kinematics controls of robotic manipulators, *Int. J. Intell. Syst. Appl.* 9 (2012) 81–91.
- [13] A.C. Nearchou, Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm, *Mech. Mach. Theory* 33 (3) (1998) 273–292.
- [14] R. Köker, A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, *Inf. Sci.* 222 (2013) 528–543.
- [15] A.A. Maciejewski, C.A. Klein, The singular value decomposition: computation and applications to robotics, *Int. J. Robot. Res.* 8 (6) (11 1989) 63–79.
- [16] R.G. Roberts, A.A. Maciejewski, Nearest optimal repeatable control strategies for kinematically redundant manipulators, *IEEE Trans. Robot. Autom.* 8 (3) (6 1992) 327–337.
- [17] B. Siciliano, The Tricept robot: Inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm, *Robotica* 17 (1999) 437–445.
- [18] H. Zhang, A parallel inverse kinematics solution for robot manipulators based on multiprocessing and linear extrapolation, *IEEE Trans. Robot. Autom.* 7 (5) (1991) 660–669.
- [19] S.R. Lucas, C.R. Tischler, A.E. Samuel, Real-time solution of the inverse kinematic-rate problem, *Int. J. Robot. Res.* 19 (2000) 1236–1244.
- [20] P. Chiacchio, S. Chiaverini, L. Sciacivco, B. Siciliano, Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority Strategy, *Int. J. Robot. Res.* 10 (1991) 410–425.
- [21] G.S. Chirikjian, J.W. Burdick, A modal approach to hyper-redundant manipulator kinematics, *IEEE Trans. Robot. Autom.* 10 (3) (1994) 343–354.
- [22] I. Ebert-Uphoff, G.S. Chirikjian, Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities, *IEEE Int Conf Robot. Autom.* 4 (1996) 139–145.
- [23] D. Sen, T.S. Mruthyunjaya, A discrete state perspective of manipulator workspaces, *Mech. Mach. Theory* 29 (4) (May 1994) 591–605.
- [24] A. Kumar, K.J. Waldron, Numerical plotting of surfaces of positioning accuracy of manipulators, *Mech. Mach. Theory* 16 (4) (1980) 361368.
- [25] Y. Wang, G.S. Chirikjian, Workspace generation of hyper-redundant manipulators as a diffusion process on SE(N), *IEEE Trans. Robot. Autom.* 20 (3) (2004) 399–408.
- [26] M. Ceccarelli, C. Lanni, A multi-objective optimum design of general 3R manipulators for prescribed workspace limits, *Mech. Mach. Theory* 39 (2004) 119–132.
- [27] I. Ebert-Uphoff, G.S. Chirikjian, Discretely actuated manipulator workspace generation by closed-form convolution, *ASME J. Mech. Des.* 120 (2) (6 1998) 245–251.
- [28] A.B. Kyatkin, G.S. Chirikjian, Computation of robot configuration and workspaces via the Fourier transform on the discrete motion group, *Int. J. Robot. Res.* 18 (6) (6 1999) 601–615.
- [29] A.B. Kyatkin, G.S. Chirikjian, Synthesis of binary manipulators using the Fourier transform on the Euclidean group, *ASME J. Mech. Des.* 121 (March 1999) 9–14.
- [30] J. Suthakorn, G.S. Chirikjian, A new inverse kinematics algorithm for binary manipulators with many actuators, *Adv. Robot.* 15 (2) (2001) 225–244.