Inverse Kinematics of Active Rotation Ball Joint Manipulators Using Workspaces Density Functions

Hui Dong, Taosha Fan, Zhijiang Du and Gregory S. Chirikjian

Abstract This paper presents a novel method to solve the inverse kinematics of redundant manipulators with active Spherical ball joint. The workspace density function is built combining the Fourier transform, convolution theorem and particular form of the workspace density of a single link, which can accurately generate the size of workspace. The approach of inverse kinematics selects a solution from among a very large discrete set using workspace density as an evaluation criterion. We then show the simulation results in the last section to proof the accuracy and precision of our method.

Keywords Redundant manipulators · Workspace density · Inverse kinematics

1 Introduction

In recent years, with the development of humanoid server robotics it has become possible to finish the mission of humans. Humanoid server robotics is demanded to move and do the action like humans. The development of active rotational joints such as the arm and neck joints which support human movement is required. There are two ways to realize the function of active rotational joints. (1) Design the special

H. Dong · Z. Du
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
e-mail: dh@hit.edu.cn; d.hui.hit@gmail.com

T. Fan · G.S. Chirikjian (✉)
Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218, USA
e-mail: gregc@jhu.edu

T. Fan
e-mail: taoshaf@gmail.com

X. Ding et al. (eds.), *Advances in Reconfigurable Mechanisms and Robots II*, Mechanisms and Machine Science 36, DOI 10.1007/978-3-319-23327-7_54

Mechanism which is drove by the general motors [1, 2]. (2) Design the spherical motor which central is the selection of compatible rotor and stator geometries [3–5].

Therefore, the inverse kinematics of active rotational joints is an important and hard topic of humanoid server robotics. Traditionally, three methods 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 [6]. The second is the algebraic model, which does not guarantee a closed-form solution, but can be efficiently solved by polynomial root finding [7]. The third is the iterative model, the result of which depends on the starting point used. The pseudo-inverse of Jacobi matrix is a typical iterative algorithm [8]. In recent years, with the development of artificial intelligence technology, neural network algorithm [9], genetic algorithm [10], fuzzy logic method [11] are widely used in solving of the inverse kinematics.

This paper presents one novel method to solve the inverse kinematics of active rotation joint based on the workspace density function of it. The workspace density functions are built by the geometric parameters of manipulators.

2 The Workspaces Density Function

General speaking, the workspace of robot is the size of the space that the end-effector of manipulators can reach. Suppose that a manipulator has N joints, and each joint has S states. The number of pose that compose the workspace of manipulator is S^N. Let the points be named the set of workspace point W that can effectively describe the workspace. The workspace W is divide into small boxes (voxels) of equal size Δx, Δy, Δz. The workspace density ρ denotes each box (voxel) of the workspace the number of points within the box that are reachable, normalized so as to be a probability density. 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 one area, the more accurately the end-effector can reach.

For every active rotation joint manipulators, there are 3-DOF that angles α, β and γ change a rotation around an any-directional axis. In Fig. 1 let a spatial rigid-body transformation be described by the homogeneous transformation

$$g(\mathbf{x}, R_{ZXZ}(\alpha, \beta, \gamma)) = \begin{pmatrix} R_{ZXZ}(\alpha, \beta, \gamma) & \mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{1}$$

where $\mathbf{x} = [x, y, z]^T$ is the translation vector described in Cartesian coordinates and α, β, γ are ZXZ Euler angles. Sometimes it will be convenient to use the shorthand $g = (\mathbf{x}, R)$ and other times it will be useful to write $g(x, y, z; \alpha, \beta, \gamma)$.

The homogeneous transformation for a single link connected to ground by a revolute joint is defined by the Denavit-Hartenberg parameters (also called DH

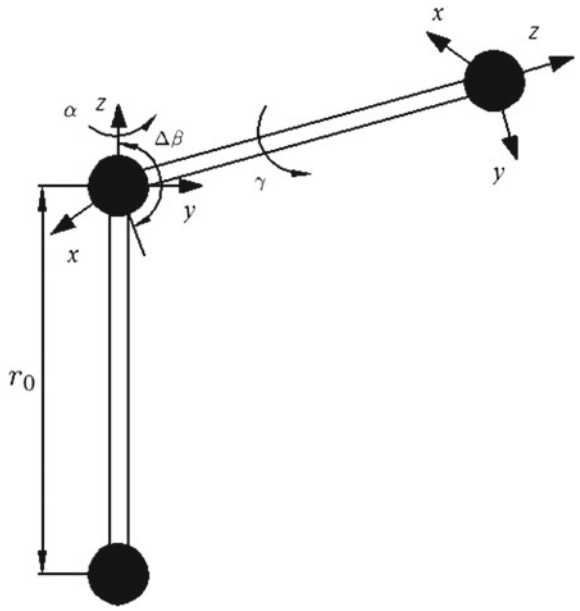

Fig. 1 Kinematic model of spatial ball joint for two links

parameters) (a, v, d, τ).[1] Where a is the length of between common normals. v the rotation angle about Z_{i-1} from X_{i-1} to X_i. d is the offset along Z_{i-1} to the common normal. τ is the angle about common normal from Z_{i-1} to Z_i.

In the Fig. 1, this places constraints on the allowable values of $x, y, z; \alpha, \beta, \gamma$. The rotation matrix of ZXZ Euler angles is denoted by the formula as

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} \mathrm{c}\alpha\mathrm{c}\beta\mathrm{c}\gamma - \mathrm{s}\alpha\mathrm{s}\gamma & -\mathrm{c}\gamma\mathrm{s}\alpha - \mathrm{c}\alpha\mathrm{c}\beta\mathrm{s}\gamma & \mathrm{c}\alpha\mathrm{s}\beta \\ \mathrm{c}\alpha\mathrm{s}\gamma + \mathrm{c}\beta\mathrm{c}\gamma\mathrm{s}\alpha & \mathrm{c}\alpha\mathrm{s}\gamma - \mathrm{c}\beta\mathrm{s}\alpha\mathrm{s}\gamma & \mathrm{s}\alpha\mathrm{s}\beta \\ -\mathrm{c}\gamma\mathrm{s}\beta & \mathrm{s}\beta\mathrm{s}\gamma & \mathrm{c}\beta \end{pmatrix} \tag{2}$$

where $\mathrm{c}\alpha = \cos\alpha$, $\mathrm{c}\beta = \cos\beta$, $\mathrm{c}\gamma = \cos\gamma$, $\mathrm{s}\alpha = \sin\alpha$, $\mathrm{s}\beta = \sin\beta$, $\mathrm{s}\gamma = \sin\gamma$.

This can be reflected in the workspace density for one link as:

$$f_1(g(\alpha, \beta, \gamma; x, y, z)) = \frac{F(\alpha)}{\sin\beta}\delta(\beta - \tau)\delta(\gamma - 0)\delta(x - a\cos v)\delta(y - a\sin v)\delta(z - d) \tag{3}$$

where $F(\alpha)$ describes the allowable range of motion of the joint around its axis. $F(\alpha)$ is constant over that range of motion, and is normalized such that $f_1(g)$ is a pdf on $SE(3)$. When the link is free to move from $[0, 2\pi)$, then $F(\alpha) = 1/2\pi$ and

[1]Note that here v is used in place of the usual notation of θ, and τ is used in place of α since both of these symbols have alternative meanings in the present formulation.

$$f_1(g(\alpha,\beta,\gamma;x,y,z)) = \frac{1}{2\pi\sin\beta}\delta(\beta-\tau)\delta(\gamma-0)\delta(x-a\cos v)\delta(y-a\sin v)\delta(z-d). \tag{4}$$

Recall that in terms of ZXZ Euler angles and Cartesian coordinates, the volume element for $SE(3)$ is

$$dg = \sin\beta d\alpha d\beta d\gamma dx dy dz. \tag{5}$$

Since IURs for $SE(3)$ are usually expressed in ZXZ Euler angles and spherical coordinates, the DH parameters are related to these coordinates by the constraints

$$\mathbf{x}(r_0,\phi_0,\theta_0) \doteq \begin{pmatrix} r_0\cos\phi_0\sin\theta_0 \\ r_0\sin\phi_0\sin\theta_0 \\ r_0\cos\theta_0 \end{pmatrix} = \begin{pmatrix} a\cos v \\ a\sin v \\ d \end{pmatrix}. \tag{6}$$

In terms of spherical coordinates,

$$\phi_0 = v \tag{7}$$

$$\theta_0 = \arctan 2(a,d) \tag{8}$$

$$r_0 = \sqrt{a^2+d^2}. \tag{9}$$

Then, instead of the Cartesian-coordinate version in (4) we can write

$$f_1(g) = \frac{1}{2\pi\sin\beta}\frac{\delta(\beta-\tau)\delta(\gamma-0)\delta(r-r_0)\delta(\theta-\theta_0)\delta(\phi-\phi_0)}{r^2\sin\theta}. \tag{10}$$

The factor $r^2\sin\theta$ results from the fact that $dxdydz = r^2\sin\theta drd\phi d\theta$.

For each value of $s \in \mathbb{Z}$ (the integers), the $SE(3)$ Fourier transform of a general probability density function $f(g)$ is

$$\hat{f}^s(p) = \int_{SE(3)} f(g)U^s(g^{-1},p)dg$$

where $U^s(g^{-1},p)$ the unitary representations of $SE(3)$. Explicitly in terms of components and coordinates

$$g = (\mathbf{a},A) = (R_{ZXZ}(\alpha,\beta,\gamma),\mathbf{x}(r,\theta,\phi)),$$

$$dg = dA\ d\mathbf{a} = (\sin\beta d\alpha d\gamma d\beta)(r^2\sin\theta drd\theta d\phi) \tag{11}$$

The element of Fourier transform for workspace density matrix is

$$\hat{f}^{s}_{l',m';l,m}(p) = \int_{\mathbb{R}^3} \int_{SO(3)} f(\mathbf{a}, A)\overline{U^{s}_{l,m;l',m'}(\mathbf{a}, A; p)}dAd\mathbf{a} \tag{12}$$

Here the fact that each $U^s(g,p)$ is an irreducible unitary representation is used, and hence

$$U^s(g^{-1}, p) = (U^s(g,p))^*$$

or in components

$$U^{s}_{l',m';l,m}((\mathbf{a}, A)^{-1}; p) = \overline{U^{s}_{l,m;l',m'}(\mathbf{a}, A; p)},$$

where the $\overline{U^{s}_{l,m;l',m'}(\mathbf{a}, A; p)}$ is the element of $U^s(g^{-1}, p)$ matrix. Explicitly,

$$\begin{aligned}\overline{U^{s}_{l,m;l',m'}(\mathbf{a}, A; p)} &= (-1)^{l-l'}(-1)^{m-m'}U^{s}_{l,-m;l',-m'}(\mathbf{a}, A; p) \\ &= (-1)^{l-l'}(-1)^{m-m'}\sum_{j=-l'}^{l'}[l, -m|p, s|l', j](\mathbf{a})\tilde{U}^{l'}_{j,-m'}(A)\end{aligned} \tag{13}$$

where

$$\begin{aligned}&[l, -m|p, s|l', j](\mathbf{a}) \\ &= (4\pi)^{\frac{1}{2}}\sum_{k=|l-l'|}^{l'+l} i^k\sqrt{\frac{(2l+1)(2k+1)}{(2l'+1)}}j_k(pr)C^{l',s}_{k,0;l,s}C^{l',j}_{k,j+m;l,-m}Y^{j+m}_{k}(\theta, \phi) \\ &\tilde{U}^{l'}_{j,-m'}(A) = (-1)^{-m'-j}\mathrm{e}^{-ij\alpha}P^{l'}_{j,-m'}(\cos\beta)\mathrm{e}^{im'\gamma}\end{aligned}$$

Here $j_k(pr)$ is kth spherical Bessel function. $Y^{j+m}_{k}(\theta, \phi)$ is spherical harmonics function. $P^{l'}_{j,-m'}(\cos\beta)$ is generalized Legendre polynomials and $C(k, j+m; l, -m|l', j)$ are Clebsch-Gordan coefficients. $P^{j+m}_{k}(\cos\theta)$ is associated Legendre function. When we substitute (10) into (12) and use (13), the result is

$$(\hat{f}_1)^{s}_{l',m';l,m}(p) = \frac{1}{2\pi}\int_0^{2\pi}\overline{U^{s}_{l,m;l',m'}(\mathbf{x}(r_0, \theta_0, \phi_0), R_{ZXZ}(v, \tau, 0); p)}dv \tag{14}$$

This integration reduces (14) to

$$(\hat{f}_1)^s_{l',m';l,m}(p) = (-1)^{l-l'}(-1)^{m-m'}\sum_{j=-l'}^{l'}(-1)^{-m'-j}P^{l'}_{j,-m'}(\cos\tau)2\pi$$
$$\delta_{0,m}(4\pi)^{\frac{1}{2}}\sum_{k=|l-l'|}^{l'+l} i^k\sqrt{\frac{(2l+1)(2k+1)}{(2l'+1)}}j_k(pr_0)$$
$$C^{l',s}_{k,0;l,s}C^{l',j}_{k,j+m;l,-m}\sqrt{\frac{(2k+1)[k-(j+m)]!}{4\pi[k+(j+m)]!}}P^{j+m}_k(\cos\theta_0) \tag{15}$$

since $\phi_0 = v$,

$$Y^{j+m}_k(\theta_0, v) = \sqrt{\frac{(2k+1)[k-(j+m)]!}{4\pi[k+(j+m)]!}}P^{j+m}_k(\cos\theta_0)\mathrm{e}^{i(j+m)v}$$

$$\int_0^{2\pi}\mathrm{e}^{-ijv}\mathrm{e}^{i(j+m)v} = 2\pi\delta_{m,0}$$

Substituting back into the general $SE(3)$ Fourier inversion formula

$$f(\mathbf{a}, A) = \frac{1}{2\pi^2}\sum_{s=-\infty}^{\infty}\sum_{l'=|s|}^{\infty}\sum_{l=|s|}^{\infty}\sum_{m'=-l'}^{l'}\sum_{m=-l}^{l}\int_0^{\infty}p^2dp\hat{f}^s_{l,m;l',m'}(p)U^s_{l',m';l,m}(\mathbf{a}, A; p), \tag{16}$$

and with the $\delta_{0,m'}$ in the expression for $(\hat{f}_1)^s_{l,m;l',m'}(p)$ gives

$$f_1(\mathbf{a}, A) = \frac{1}{2\pi^2}\sum_{s=-\infty}^{\infty}\sum_{l'=|s|}^{\infty}\sum_{l=|s|}^{\infty}\sum_{m'=-l'}^{l'}\int_0^{\infty}p^2dp\hat{f}^s_{l,m;l',0}(p)U^s_{l',0;l,m}(\mathbf{a}, A; p) \tag{17}$$

One of the most important properties of the Fourier transform of functions on $\mathbb{R}^N$ is that the Fourier transform of the convolution of two functions is the product of the Fourier transform of the functions. This property persists also for the convolution of functions on the group, namely

$$\begin{aligned}\mathscr{F}(f_1 * f_2) &= \int_G(\int_G f_2(g)U(g^{-1}, p)U(h^{-1}, p)d(g))f_1(h)d(h)\\ &= (\int_G f_2(g)U(g^{-1}, p)d(g))(\int_G f_1(h)U(h^{-1}, p)d(h))\\ &= \mathscr{F}(f_2)\mathscr{F}(f_1) = \hat{f}_2(p)\hat{f}_1(p)\end{aligned} \tag{18}$$

What we are really interested in is the n-fold convolution

$$f_n(g) = (f_1 * f_1 * \cdots f_1)(g),$$

and even more general than that, the case when the links are all different. But for now we consider the case when they are all the same.

Note that since $\delta_{0,m'} \cdot \delta_{0,m'} = \delta_{0,m'}$, we can write

$$(\hat{f}_1)^s_{l,m;l',m'}(p) = (\hat{f}_1)^s_{l,m;l',m'}(p)\delta_{0,m}.$$

Then, for each fixed value of s and p,

$$\begin{aligned}(\hat{f}_2)^s_{l,m;l'',m''}(p) &= \sum_{l'=|s|}^{\infty}\sum_{m'=-l'}^{l'}(\hat{f}_1)^s_{l,m;l',m'}(p)(\hat{f}_1)^s_{l',m';l'',m''}(p)\\ &= \sum_{l'=|s|}^{\infty}\sum_{m'=-l'}^{l'}(\hat{f}_1)^s_{l,m;l',m'}(p)\,\delta_{0,m'}\,(\hat{f}_1)^s_{l',m';l'',m''}(p)\,\delta_{0,m''} \qquad (19)\\ &= \delta_{0,m''}\sum_{l'=|s|}^{\infty}(\hat{f}_1)^s_{l,m;l',0}(p)(\hat{f}_1)^s_{l',0;l'',m''}(p).\end{aligned}$$

This is significant for two reasons: (1) one of the sums disappeared is going from (19); (2) because now we can go back to (16), and evaluate at $m' = 0$ to simplify as

$$\begin{aligned}(\hat{f}_n)^s_{l,m;l'',m''}(p) &= \sum_{l'=|s|}^{\infty}\sum_{m'=-l'}^{l'}(\hat{f}_{n-1})^s_{l,m;l',m'}(p)(\hat{f}_1)^s_{l',m';l'',m''}(p)\\ &= \sum_{l'=|s|}^{\infty}\sum_{m'=-l'}^{l'}(\hat{f}_{n-1})^s_{l,m;l',m'}(p)\delta_{0,m'}(\hat{f}_1)^s_{l',m';l'',m''}(p)\delta_{0,m''}\\ &= \delta_{0,m''}\sum_{l'=|s|}^{\infty}(\hat{f}_{n-1})^s_{l,m;l',0}(p)(\hat{f}_1)^s_{l',0;l'',m''}(p).\end{aligned}$$

Substituting back into the general $SE(3)$ Fourier inversion formula for n links.

$$f_n(\mathbf{a},A) = \frac{1}{2\pi^2}\sum_{s=-\infty}^{\infty}\sum_{l''=|s|}^{\infty}\sum_{l=|s|}^{\infty}\int_0^{\infty}p^2dp(\hat{f}_n)^s_{l,m;l'',0}(p)U^s_{l'',0;l,m}(\mathbf{a},A;p) \qquad (20)$$

3 Simplify the Workspace Density Function of Spherical Hinge Manipulators

According to previous work, the workspace density of spherical hinge is

$$f(R(\alpha,\beta,\gamma),\mathbf{x}(r,\phi,\theta)) = C\cdot\frac{\delta(r-L)\delta(\alpha-\phi)\delta(\beta-\theta)}{r^2\sin\theta} \tag{21}$$

where C is a constant determined by motion range of the spherical hinge.

Note $A\in SO(3)$ is distance and orientation preserved, thus the relative distance, $r-L$, and the relative angles, $\alpha-\phi$ and $\beta-\theta$, are also preserved under A. Then $f(R,\mathbf{x})$ remains unchanged under $SO(3)$, i.e.

$$\forall A\in SO(3), f(R,\mathbf{x}) = f(AR,A\mathbf{x}) \tag{22}$$

For a robot arm of N links, the probability density of pose $g(R,x)$ is

$$f_0^N(R,\mathbf{x}) = \int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{y})f_{N-1}^N(A^TR,A^T(\mathbf{x}-\mathbf{y}))dAd\mathbf{y} \tag{23}$$

Make coordinate change by $\mathbf{y}=\mathbf{x}-\mathbf{y}'$,

$$\begin{aligned} f_0^N(R,\mathbf{x}) &= \int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{y})f_{N-1}^N(A^TR,A^T(\mathbf{x}-\mathbf{y}))dAd\mathbf{y} \\ &= \int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{x}-\mathbf{y}')f_{N-1}^N(A^TR(a,b,g),A^T\mathbf{y}'(r',\phi',\theta'))dAd\mathbf{y}' \end{aligned} \tag{24}$$

Substitute (21) and (22) to (24), we can get

$$\begin{aligned} &f_0^N(R(\alpha,\beta,\gamma),\mathbf{x}(r,\phi,\theta)) \\ &= \int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{x}-\mathbf{y}')f_{N-1}^N(A^TR(\alpha,\beta,\gamma),A^T\mathbf{y}'(r',\phi',\theta'))dAd\mathbf{y}' \\ &= C\cdot\int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{x}-\mathbf{y}')\frac{\delta(r'-L)\delta(\alpha-\phi')\delta(\beta-\theta')}{r'^2\sin\theta}\cdot r'^2\sin\theta dAdr'd\theta'd\phi' \\ &= C\cdot\int\limits_{SO(3)}\int\limits_{\mathbb{R}^3} f_0^{N-1}(A,\mathbf{x}-\mathbf{y}'(L,\alpha,\beta))dA \end{aligned} \tag{25}$$

As for $\mathbf{x} - \mathbf{y}'$,

$$\mathbf{x} - \mathbf{y}'(L, \alpha, \beta) = R(\alpha, \beta, \gamma) \begin{bmatrix} 0 \\ 0 \\ -L \end{bmatrix} + \mathbf{x} = g(R(\alpha, \beta, \gamma), \mathbf{x}) \begin{bmatrix} 0 \\ 0 \\ -L \end{bmatrix} \tag{26}$$

Let $y'' = x - y'(L, \alpha, \beta)$ and substitute it to (25),

$$f_0^N(R(\alpha, \beta, \gamma), \mathbf{x}(r, \phi, \theta)) = C \cdot \int_{SO(3)} \int_{\mathbb{R}^3} f_0^{N-1}(A, \mathbf{y}'') dA|_{y''=g(R,\mathbf{x})[0\ 0\ -L]^T} \tag{27}$$

where $\int_{SO(3)} \int_{\mathbb{R}^3} f_0^{N-1}(A, \mathbf{y}'') dA$ is the probability of the first N − 1 links reaches the desired point $\mathbf{y}''$.

This simplification could make calculation of the probability density function easier. For example, for a manipulator consisted of N spherical joints, the probability density function of a desired pose without simplification is

$$f_0^N(R, \mathbf{x}) = \frac{1}{2\pi^2} \sum_{s=-\infty}^{\infty} \sum_{l=|s|}^{\infty} \sum_{l'=|s|}^{\infty} \int_0^{\infty} p^2 dp\, f_{l,0,l',0}(p) U_{l',0,l,0}(R, \mathbf{x}, p) \tag{28}$$

whereas, the probability density function of desired point is

$$f_0^N(\mathbf{x}(r, \phi, \theta)) = \int_{SO(3)} f_0^N(A, \mathbf{x}) dA = \frac{1}{2\pi^2} \int_0^{\infty} p^2 dp \prod_{i=1}^{N} j_0(pL_i) j_0(pr). \tag{29}$$

4 Inverse Kinematics

A method is proposed to solve the inverse kinematics based on Fourier-based workspace density which is similar to [12] which presented an inverse kinematics algorithm for planar serial revolute manipulators. The criterion of this method is to select parameters of ball joint so as to obtain the maximum workspace density to fix the configuration of the manipulator. The point which generating maximum of workspace density function express the best flexibility of the manipulator at that point.

Consider one actuated rotation joint manipulator with N links. g_k is denoted the homogenous transformation matrix of the kth link which from its own base to the distal end, where k $\in$ {1, 2,…, N}. The homogenous transformation matrix that from the base point to the end of kth link is denoted $g^{(k)}$.

$$g^{(k)} = g_1 \circ g_2 \circ \cdots \circ g_k \tag{30}$$

The homogenous transformation matrix that from the kth link to the distal end of the manipulator is

$$(g^{(k)})^{-1} \circ g^{(N)} = g_{k+1} \circ g_{k+2} \circ \cdots \circ g_N \tag{31}$$

In the Fig. 2, g_{aim} is denoted the target point. All the possible states of one module is computed and put into a set of $G = \{g_{ii}\}$. For the kth link, where $k < N$, to find g_k which makes maximizes the workspace density $f_{N-k}((g_1^* \circ g_2^* \circ \cdots \circ g_k)^{-1} \circ g_{aim})$. Then we fix the transformation g_k^* for the kth link. Then we proceed up the manipulator one link. Among all possible states, we search for the state of g_{k+1} that can achieve the highest density $f_{N-k-1}((g_1^* \circ g_2^* \circ \cdots \circ g_k^* \circ g_{k+1})^{-1} \circ g_{aim})$. If g_{k+1} is such a state, we configure the $(k+1)$th link to g_{k+1}^*. When $k = N$, we fix the g_N which makes minimizes the

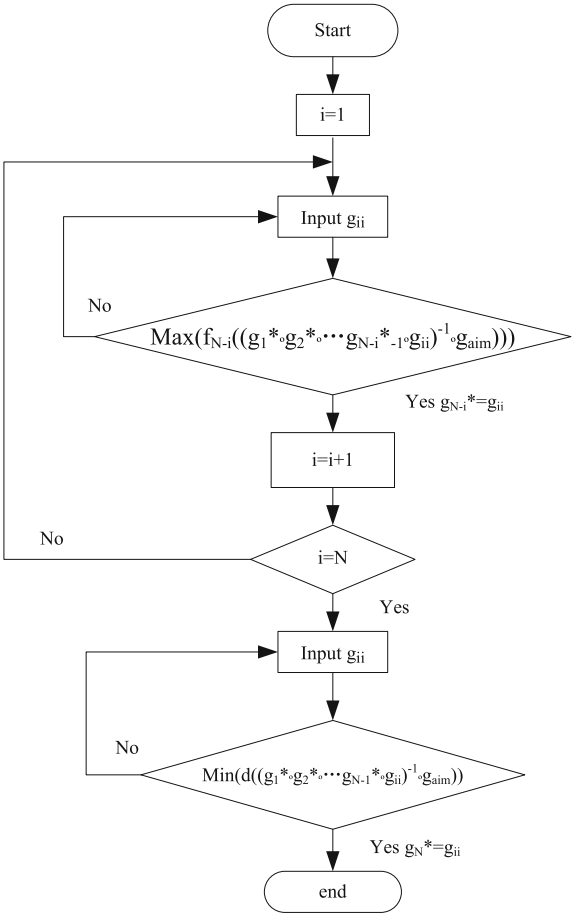

Fig. 2 Flow chart of inverse kinematics algorithm

distance between the distal end of manipulator and the aim pose $d_{N-k}((g_1^* \circ g_2^* \circ \cdots \circ g_k)^{-1} \circ g_{aim})$.

5 Numerical Simulations for Inverse Kinematics

To illustrate the usefulness of the inverse kinematics algorithm based on workspace density functions, we solve the inverse kinematics of active Rotation joint manipulators with different numbers of links. The simulation result of inverse kinematics method is shown in Figs. 3 and 4.

In the Fig. 3, the length of each link $L = 1$, the number of links $N = 8$. In the Fig. 2, the target pose $(x_{aim}, y_{aim}, z_{aim}) = (3.5, 4, 5)$. The target pose $(x_{aim}, y_{aim}, z_{aim}) = (-4, -3, -5)$ in the Fig. 2. The inverse kinematics simulation results of 6-link are shown in Fig. 4, where the length of each link $L = 1$, the number of links $N = 6$. The target pose $(x_{aim}, y_{aim}, z_{aim}) = (-40, -3.5, 15)$ and $(x_{aim}, y_{aim}, z_{aim}) = (3, 1, 5)$ are computed in the Fig. 4a, b.

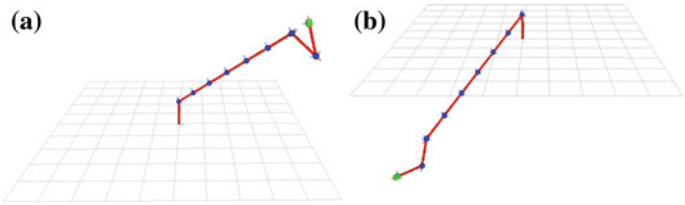

Fig. 3 The inverse kinematics simulation results of 8-link manipulator

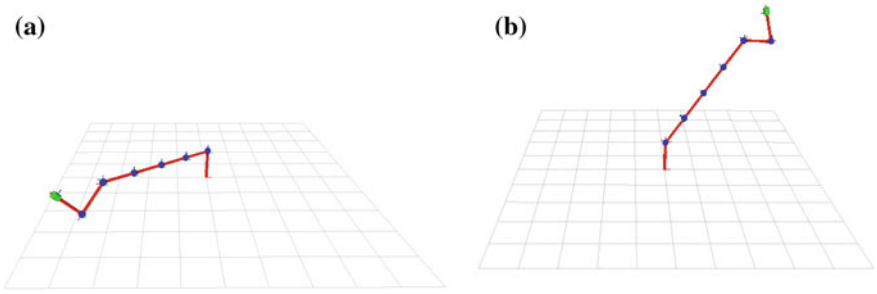

Fig. 4 The inverse kinematics simulation results of 6-link manipulator

6 Conclusion

Combining the concept of the Fourier transform for the group of rigid-body motion of $SE(3)$, the convolution theorem and the workspace density functions of active rotation ball joint, we propose the inverse kinematics algorithm can efficiently computed. 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 functions as an evaluation criterion.

Acknowledgments Chirikjians contribution to this material is based upon work supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Yu, Y., Narita, Y., Harada, Y., et al.: Development of 3-DOF active rotational ball joint. In: IEEE International Conference on Robotics and Biomimetics, 2007. ROBIO 2007. IEEE, New York, pp. 1966–1971 (2007)
2. Lee, K.M., Ezenekwe, D.E., He, T.: Design and control of a spherical air-bearing system for multi-dof ball-joint-like actuators. Mechatronics **13**(2), 175–194 (2002)
3. Wang, W., Wang, J., Jewell, G.W., et al.: Design and control of a novel spherical permanent magnet actuator with three degrees of freedom. IEEE/ASME Trans. Mechatron. **8**(4), 457–468 (2003)
4. Chirikjian, G.S., Stein, D.: Kinematic design and commutation of a spherical stepper motor. IEEE/ASME Trans. Mechatron. **4**(4), 342–353 (1999)
5. Yano, T., Suzuki, T.: Basic characteristics of the small spherical stepping motor. In: Proceedings of the IEEE International Conference Intelligent Robots and Systems, pp. 1980–1985 (2002)
6. Perez, A., McCarthy, J.M.: Sizing a serial chain to fit a task trajectory using Clifford algebra exponentials. In: IEEE International Conference on Robotics and Automation (ICRA 2005), pp. 4709–4715 (2005)
7. Yahya, S., Moghavvemi, M., Yang, S.S., Mohamed, H.A.F.: Motion planning of hyper redundant manipulators based on a new geometrical method. In: IEEE International Conference on Industrial Technology (ICIT 2009), Australia, pp. 1–5 (2009)
8. Dulęba, I., Opałka, M.: A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators. Int. J. Appl. Math. Comput. Sci. **23**(2), 373–382 (2013)
9. Daachi, B., Benallegue, A.: A neural network adaptive controller for end-effector tracking of redundant robot manipulators. J. Intell. Rob. Syst. **46**(3), 245–262 (2006)
10. Nearchou, A.C.: Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm. Mech. Mach. Theory **33**(3), 273–292 (1998)
11. Kubota, N., Arakawa, T., Fukada, T.: Motion learning for redundant manipulator with structured intelligence. In: Annual Conference of the IEEE Industrial Electronics Society (IECON98), vol. 1, pp. 104–109 (1998)
12. Dong, H., Du, Z., Chirikjian, G.S.: Workspace density and inverse kinematics for planar serial revolute manipulators. Mech. Mach. Theory **70**, 508–522 (2013)