

A new inverse kinematics algorithm for binary manipulators with many actuators

JACKRIT SUTHAKORN and GREGORY S. CHIRIKJIAN*

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

Received 17 May 2000; accepted 29 July 2000

Abstract—In this paper we present a new, and extremely fast, algorithm for the inverse kinematics of discretely actuated manipulator arms with many degrees of freedom. Our only assumption is that the arm is macroscopically serial in structure, meaning that the overall structure is a serial cascade of units with each unit having either a serial or parallel kinematic structure. Our algorithm builds on previous works in which the authors and coworkers have used the workspace density function in a breadth-first search for solving the inverse kinematics problem. The novelty of the method presented here is that only the ‘mean’ of this workspace density function is used. Hence the requirement of storing a sampled version of the workspace density function (which is a function on a six-dimensional space in the case of a spatial manipulator) is circumvented. We illustrate the technique with both planar revolute and variable-geometry-truss manipulators, and briefly describe a new manipulator design for which this algorithm is applicable.

Keywords: Inverse kinematics; discrete actuation; rigid-body motion; probability density function; statistics; groups.

1. INTRODUCTION

In this section we present a brief introduction to our method and review the literature.

1.1. Overview of our approach

In this paper we present a new algorithm for the inverse kinematics of discretely actuated manipulator arms with many degrees of freedom. Our approach is extremely fast and requires minimal data storage. Our only assumption is that the discretely actuated arm is macroscopically serial in structure, meaning that the overall structure is a serial cascade of units with each unit having either a serial

*To whom correspondence should be addressed. E-mail: gregc@jhu.edu

or parallel kinematic structure. Our algorithm builds on previous works in which the authors and coworkers have used the workspace density function in a breadth-first search for solving the inverse kinematics problem. The workspace density function is a probability density function (pdf) on the group of rigid-body motions (this group is often called $SE(3)$ in the robotics literature [1, 2]). The workspace density is an important quantity in the design and motion planning of discretely actuated manipulators in analogy with the way dexterity measures are used in the context of continuous-motion manipulators. See [3–6] for discussions of dexterity measures.

The novelty of the method presented here is that only the ‘mean’ of this workspace density function is stored and used. This leads to a tremendous reduction in the amount of data that needs to be stored. However, in order to fully understand our method, knowledge of some definitions used in the field of probability and statistics on groups such as the mean and variance of pdfs is required (see, e.g. [7]). We therefore review these definitions in this paper and perform concrete statistical calculations that we believe to be new.

1.2. Literature review

The concept of discretely actuated manipulators is quite old in the literature. A planar serial revolute ‘digital manipulator’ is discussed in Pieper’s classic thesis from 1968 [8]. In another classic paper, Roth *et al.* discuss a three-dimensional (3D) digital manipulator actuated with inflatable airbags [9]. In the mid 1980s, discretely actuated manipulator arms were developed in the former Soviet Union by Koliskor [10]. In fact, a low degree-of-freedom discrete-state robot has been sold by Seiko for pick-and-place tasks [11]. Closely related to the concept of a discretely actuated manipulator is the idea of sampling a continuous-motion robot at discrete values. This has been done to analyze the error in robotic mechanisms and to generate their workspaces using the Monte Carlo method. See, e.g. [12, 13].

Since the mid 1990s Chirikjian and coworkers have developed a variety of efficient algorithms for highly actuated discrete-state robots and mechanisms. These include approaches to the kinematic synthesis of such mechanisms [14–17], the generation of workspaces [18, 19] and inverse kinematics [20–25]. Figure 1 shows a 3D discrete-state manipulator arm with 2^{36} states. This arm consists of six Stewart/Gough platforms actuated with pneumatic cylinders. This type of arm (or even one with twice as many actuators and 2^{72} states) provides motivation for our exploration of efficient inverse kinematics algorithms.

In the subsequent sections of this paper, we present a new approach to the inverse kinematics of discrete-state manipulators. Section 2 reviews and extends the Ebert-Uphoff algorithm presented in [25]. Our new algorithm builds on this. Since our algorithm relies on probability theory, we review concepts of basic probability in Section 3 and present some new results in probability theory applied to the group of rigid-body motions in Section 4. We then use these ideas to formulate our algorithm in Section 5 and illustrate our algorithm with numerical examples in Section 6.

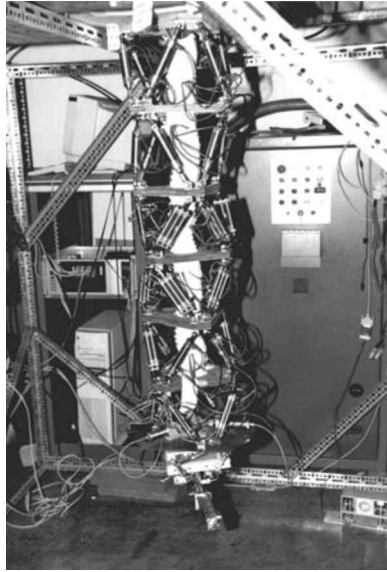


Figure 1. A 2^{36} -state hybrid serial–parallel manipulator arm.

2. INVERSE KINEMATICS OF BINARY MANIPULATORS: THE EXTENDED EBERT-UPHOFF ALGORITHM

In this section we review and reformulate an algorithm that was presented in [25] for solving the positional inverse kinematics problem for discretely actuated manipulators, and extend it to the combined positional and orientational problem.

Suppose it were possible to explicitly enumerate all possible end positions and orientations reachable by a discretely actuated manipulator arm. This finite set of frames could then be approximated as a histogram on the 6D space of positions and orientations. This space is simply $SE(3)$, the group of rigid-body motions. Likewise, the group of rigid-body motions of the plane is called $SE(2)$.

Let an arbitrary element of $SE(3)$ be denoted as:

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

where A is a 3×3 rotation matrix (element of $SO(3)$) and $\mathbf{a} \in \mathbb{R}^3$ is a translation in 3D space. Then we can describe a normalized histogram as a pdf on $SE(3)$. Henceforth we use the notations G and $SE(3)$ interchangeably (G standing for 'group'). In general, a pdf on a group satisfies the conditions $\rho(g) \geq 0$ and:

$$\int_G \rho(g) dg = 1, \quad (2)$$

for an appropriate integration measure dg . In the case of $SE(3)$, $dg = dA d\mathbf{a}$ where $d\mathbf{a} = da_1 da_2 da_3$ and $dA = \sin \beta d\alpha d\beta d\gamma$ when ZXZ Euler angles are used to parameterize rotations. That is, the form of dA depends on the parameterization

used in much the same way that \mathbf{da} would appear differently if polar or spherical coordinates were used. Explicitly, (2) is written as:

$$\int_{a_3=-\infty}^{\infty} \int_{a_2=-\infty}^{\infty} \int_{a_1=-\infty}^{\infty} \int_{\gamma=0}^{2\pi} \int_{\beta=0}^{\pi} \int_{\alpha=0}^{2\pi} \rho(\alpha, \beta, \gamma, a_1, a_2, a_3) \\ \times \sin \beta \, d\alpha \, d\beta \, d\gamma \, da_1 \, da_2 \, da_3 = 1.$$

If the manipulator consists of P not-necessarily-identical modules, we denote the pdf for the whole manipulator as $\rho_{0,P}(g)$. We denote the pdf for the i th module simply as $\rho_{i-1,i}$. It was shown in [19] that the pdf for two concatenated modules can be found as the convolution:

$$\rho_{i-1,i+1}(g) = (\rho_{i-1,i} * \rho_{i,i+1})(g) \triangleq \int_G \rho_{i-1,i}(h) \rho_{i,i+1}(h^{-1} \circ g) \, dh.$$

By the same reasoning, it follows that the pdf for the whole manipulator can be found as:

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

The concept of the manipulator workspace density functions $\rho_{i,j}(g)$ for $i < j$ is useful in solving the inverse kinematics problem for discretely actuated manipulators with many states [25, 16]. If one were to try to solve the inverse kinematics problem by evaluating the manipulator forward kinematics for a concatenation of P identical modules each with K states, the computational cost of explicitly searching K^P possibilities would be prohibitive. If we have a cascade of manipulator workspace density functions corresponding to each of the P modules of the manipulator, this exponential complexity can be reduced to a problem that is linear in P .

Now for $k \in \{1, \dots, P\}$, let g_k be the transformation that relates the distal end of the k th segment to its base for a given configuration of the module. The position of the top of the k th segment relative to the base of the manipulator is then:

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

and the position and orientation of the distal end of the manipulator relative to the distal end of the k th segment is

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

There are K possible states for each g_k . Using the information in the cascade of density functions $\rho_{P-1,P}, \dots, \rho_{1,P}$, we can sequentially choose states of each section which, at each instant, maximize the probability density around the particular frame of reference that we seek to reach.

In other words, given that we want the end of the manipulator to reach $g_{\text{des}} \in G$, we start at the base of the manipulator and ask which state of segment 1 maximizes $\rho_{1,P}((g^{(1)})^{-1} \circ g_{\text{des}})$. After searching through all K possible values of g_1 and the optimal $g^{(1)} = g_1$ is fixed, we proceed up the manipulator one unit. That is, we next calculate $\rho_{2,P}((g^{(2)})^{-1} \circ g_{\text{des}})$. Since g_1 is fixed, K values of $g^{(2)} = g_1 \circ g_2$ are

searched until the value of g_2 that maximizes is found. This procedure is performed by sequentially maximizing $\rho_{k,P}((g^{(k)})^{-1} \circ g_{\text{des}})$ for all $k \in \{1, 2, \dots, P-1\}$. When $k = P$, the one out of K values of g_P that minimizes the cost function

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

is chosen where $D(\cdot, \cdot)$ is a distance metric. Such metrics for rigid-body motions have been discussed in [26–30].

The procedure we have described here is a way of specifying a state of the whole manipulator in $\mathcal{O}(P)$ arithmetic operations such that the distal end reaches g_{des} approximately. The spirit of this algorithm is the same as that presented in [25], though the use of $SE(3)$ distance metrics and other group-theoretic features allows us to consider the full position-orientation inverse kinematics problem rather than the pure position problem. Hence, we call this the extended Ebert-Uphoff algorithm.

3. REVIEW OF BASIC PROBABILITY

One drawback of the algorithm described in the previous section is that it requires us to store $\rho_{k,P}(g)$ for $k = 0, \dots, P-1$. Each of these is a function on a 6D space, which means that even though the algorithm is very fast, a tremendous amount of data must be stored. For instance, if G is sampled at 100 values in each parameter, $P \times 10^{12}$ numbers must be stored. This has lead us to consider a modification of the extended Ebert-Uphoff algorithm in which only the mean values of the pdfs $\rho_{k,P}(g)$ for $k = 0, \dots, P-1$ need be stored. That is, we store only P homogeneous transforms. The difficulty comes in finding a correct and useful definition of the mean value of a function on a group. We do this in the next section, and present background here that enables that formulation to make sense to non-mathematicians.

Given a pdf with a real argument, $\rho(x)$, the classical apparatus of probability theory is used to analyze $\rho(x)$ in terms of its moments. The expected value is the center of mass, or mean, $E(x) = x_{\text{cm}}$ defined as the solution of:

$$\int_R (x - x_{\text{cm}})\rho(x) dx = 0.$$

The n th moments about the expected value $E(x) = x_{\text{cm}}$ for all integers $n \geq 0$ are given by:

$$M_n = \int_R (x - x_{\text{cm}})^n \rho(x) dx.$$

$M_0 = 1$ is the ‘mass’ of the pdf and $M_1 = 0$ by definition. $M_2 = \sigma^2$ is the variance, where σ is the standard deviation. Knowing all of the moments from $n = 2, \dots, N$ for some finite N means that the important properties of the distribution $\rho(x)$ are known to some degree without regard to all of its details. As $N \rightarrow \infty$, the properties

of $\rho(x)$ become more determined as more of the moments are known. Usually only the first few moments are of concern.

It is easy to show by direct calculation that $E(x) = x_{\text{cm}}$ may be calculated as the value of y which minimizes the functional:

$$C(y) = \int_R (x - y)^2 \rho(x) dx. \quad (3)$$

Hence, the value of y that minimizes $C(y)$ is the mean and $C(x_{\text{cm}})$, the function C evaluated at the mean, is the variance.

The moments of pdfs have nice properties under the operation of convolution. Recall that on the line:

$$(\rho_1 * \rho_2)(x) = \int_{-\infty}^{\infty} \rho_1(\xi) \rho_2(x - \xi) d\xi.$$

Since $\rho_1(x) \geq 0$ for all $x \in \mathbb{R}$, it follows from a change of variables $z = x - \xi$ that the mass of the convolution of two pdfs is unity:

$$\int_{-\infty}^{\infty} (\rho_1 * \rho_2)(x) dx = \left(\int_{-\infty}^{\infty} \rho_1(\xi) d\xi \right) \left(\int_{-\infty}^{\infty} \rho_2(z) dz \right) = 1 \cdot 1 = 1. \quad (4)$$

That is, the convolution of two pdfs results in a pdf. Similarly, one has that the higher moments of convolutions of pdfs can be generated from the moments of the pdfs being convolved:

$$x_{\text{cm}}^{1*2} = \int_{-\infty}^{\infty} x (\rho_1 * \rho_2)(x) dx = \int_{-\infty}^{\infty} x \rho_1(x) dx + \int_{-\infty}^{\infty} x \rho_2(x) dx = x_{\text{cm}}^1 + x_{\text{cm}}^2, \quad (5)$$

and

$$\begin{aligned} (\sigma^2)_{1*2} &= \int_{-\infty}^{\infty} (x - x_{\text{cm}}^{1*2})^2 (\rho_1 * \rho_2)(x) dx \\ &= \int_{-\infty}^{\infty} (x - x_{\text{cm}}^1)^2 \rho_1(x) dx + \int_{-\infty}^{\infty} (x - x_{\text{cm}}^2)^2 \rho_2(x) dx \\ &= (\sigma^2)_1 + (\sigma^2)_2. \end{aligned} \quad (6)$$

As with (4), we find (5) and (6) by substituting $z = x - \xi$ and using the invariance of integration under shifts.

4. MEAN AND VARIANCE FOR $SO \cdot N/$ AND $SE \cdot N/$

We now demonstrate the definitions of mean and variance in the context of $G = SE(N)$ for $N = 2$ and 3 . For $SE(N)$ there are a variety of metrics we can use (see, e.g. [31], chapters 5 and 6). Here we use the metric:

$$D(g_1, g_2) = \sqrt{\|\mathbf{a}_1 - \mathbf{a}_2\|_2^2 + L^2 \|A_1 - A_2\|_2^2},$$

where $g_i = (\mathbf{a}_i, A_i) \in SE(3)$ and $L \in \mathbb{R}^+$ is a length scale to put orientational and positional quantities in the same units. The mean of a pdf $\rho(g)$ is then defined to be the value $g_{\text{cm}} \in G$ such that:

$$C(g) = \int_G D^2(g, h)\rho(h) dh,$$

is minimized.

In the following subsections, we examine the explicit form of g_{cm} for any given pdf and show some interesting properties analogous to (5) and (6). Our approach has some similarities with [32].

4.1. Explicit calculation for $SO(N)$

It may be shown that finding g_{cm} amounts to finding an $SO(N)$ mean and an \mathbb{R}^N mean. Since the \mathbb{R}^N case is what is addressed in classical probability and statistics, we take some time here to consider the $SO(N)$ case. The two results are combined in the next subsection.

Given a pdf $\rho \in \mathcal{L}^2(SO(N))$, our goal is to find $A_1 \in SO(N)$ that minimizes the function:

$$C(A_1) = \int_{SO(N)} \|A_1 - A_2\|_2^2 \rho(A_2) dA_2.$$

The constrained minimization problem of finding this value of A_1 (which defines A_{cm}) is almost identical to the problem of finding a rotation matrix that best fits two sets of corresponding points. Such problems have been studied extensively in image analysis and computer vision [33–43], and spacecraft attitude determination [44–49].

As it turns out, the answer to the constrained problem is to first minimize without constraint and select the rotation matrix from the polar decomposition of the answer to the unconstrained problem. It is this rotation that will be the solution to the constrained problem. Explicitly, this means that we first compute:

$$M = \int_{SO(N)} A\rho(A) dA.$$

The matrix M in general is not a rotation matrix even though A is. To get the closest rotation to M (which happens to be the solution to the constrained minimization of $C(A)$), we calculate

$$A_{\text{cm}} = M(M^T M)^{(-1/2)}. \tag{7}$$

An interesting feature of this solution is that it has some similarities to the case of the real line. For instance, if we have calculated M_1 and M_2 corresponding to pdfs $\rho_1(A)$ and $\rho_2(A)$, then:

$$A_{\text{cm}}^{1*2} = M_{1*2}(M_{1*2}^T M_{1*2})^{(-1/2)},$$

where

$$M_{1*2} = M_1 M_2.$$

Note, however, that for the 3D case:

$$A_{\text{cm}}^{1*2} \neq A_{\text{cm}}^1 A_{\text{cm}}^2.$$

To see that this is true, we must use the invariance properties of the integration measure dA . Namely:

$$\int_{SO(N)} A(\rho_1 * \rho_2)(A) dA = \int_{SO(N)} A \int_{SO(N)} \rho_1(R) \rho_2(R^T A) dR dA,$$

can be rewritten using the change of variables $Q = R^T A$ as

$$\begin{aligned} & \int_{SO(N)} RQ \int_{SO(N)} \rho_1(R) \rho_2(Q) dQ dR \\ &= \left(\int_{SO(N)} R \rho_1(R) dR \right) \left(\int_{SO(N)} Q \rho_2(Q) dQ \right) = M_1 M_2. \end{aligned}$$

What this means is that we do not explicitly need to know $(\rho_1 * \rho_2)(A)$ to calculate A_{cm}^{1*2} . The impact that this has on us being able to do inverse kinematics efficiently is that we can calculate the orientational mean for each section of the manipulator individually and circumvent the convolution procedure. This observation will be used in the next section.

4.2. Explicit calculation for $SE(2)$ and $SE(3)$

The mean (expected value) associated with a pdf $\rho \in \mathcal{L}^2(SE(N))$ is the pair $(\mathbf{a}_{\text{cm}}, A_{\text{cm}}) \in SE(N)$ that minimizes the function:

$$C(\mathbf{a}_1, A_1) = \int_{R^N} \int_{SO(N)} \{ \|\mathbf{a}_1 - \mathbf{a}_2\|_2^2 + L \|A_1 - A_2\|_2^2 \} \rho(\mathbf{a}_2, A_2) d\mathbf{a}_2 dA_2,$$

subject to the constraint that $A_1 \in SO(N)$. The minimization with respect to \mathbf{a}_1 follows exactly like the case of a function on \mathbb{R}^N and we find the value to be

$$\mathbf{a}_{\text{cm}} = \int_{R^N} \mathbf{a}_2 \left(\int_{SO(N)} \rho(\mathbf{a}_2, A_2) dA_2 \right) d\mathbf{a}_2. \quad (8)$$

In order to minimize with respect to A_1 , the results of the previous subsection concerning $SO(N)$ exactly apply. The only difference is that now:

$$M = \int_{SO(N)} A \left(\int_{R^N} \rho(\mathbf{a}, A) d\mathbf{a} \right) dA.$$

As with the case of $SO(N)$, we know of no closed-form solution for the orientational means and variances of the convolution of two pdfs on $SE(N)$ in terms of strictly the means and variances of the original pdfs (since we cannot recover M

uniquely from A_{cm}). However, we are able to write the translational mean of the convolution of two pdfs in terms of the translational means of each pdf. In order to see this, evaluate (8) with the pdf $(\rho_1 * \rho_2)(\mathbf{a}, A)$. We denote the result as:

$$\mathbf{a}_{\text{cm}}^{1*2} = \int_{R^N} \mathbf{a} \left(\int_{SO(N)} (\rho_1 * \rho_2)(\mathbf{a}, A) dA \right) d\mathbf{a}.$$

Substitution of the explicit form of $(\rho_1 * \rho_2)(\mathbf{a}, A)$ into the above equation yields:

$$\mathbf{a}_{\text{cm}}^{1*2} = \int_{R^N} \mathbf{a} \int_{SE(N)} \rho_1(\mathbf{r}, R) \left(\int_{SO(N)} \rho_2(R^{-1}(\mathbf{a} - \mathbf{r}), R^{-1}A) dA \right) d\mathbf{r} dR d\mathbf{a}.$$

Due to the invariance of integration on $SO(N)$, we may simplify the quantity inside the parenthesis by defining:

$$F_2(\mathbf{a}) \triangleq \int_{SO(N)} \rho_2(\mathbf{a}, A) dA.$$

Making the substitution $\mathbf{x} = R^{-1}(\mathbf{a} - \mathbf{r})$, we then have:

$$\mathbf{a}_{\text{cm}}^{1*2} = \int_{SE(N)} \int_{R^N} (R\mathbf{x} + \mathbf{r}) \rho_1(\mathbf{r}, R) F_2(\mathbf{x}) d\mathbf{r} dR d\mathbf{x}.$$

Passing integrals through terms that are invariant under the integral and using the fact that:

$$\int_{SE(N)} \rho_i(g) dg = 1,$$

we find:

$$\mathbf{a}_{\text{cm}}^{1*2} = \mathbf{a}_{\text{cm}}^1 + M_1 \mathbf{a}_{\text{cm}}^2, \tag{9}$$

where \mathbf{a}_{cm}^i is the translational part of the mean of ρ_i and

$$M_1 = \int_{SO(N)} R \left(\int_{R^N} \rho_1(\mathbf{r}, R) d\mathbf{r} \right) dR.$$

As an example of the usefulness of (9), consider the mean of the convolution of a pdf $\rho(\mathbf{a}, A)$ with itself P times. If \mathbf{a}_{cm} is the translational mean of $\rho(\mathbf{a}, A)$, then the translational mean of $(\rho * \rho * \dots * \rho)(\mathbf{a}, A)$ will be

$$\mathbf{a}_{\text{cm}}^{1*\dots*P} = \left(\mathbf{1} + \sum_{k=1}^{P-1} M^k \right) \mathbf{a}_{\text{cm}}.$$

However, since:

$$\left(\mathbf{1} + \sum_{k=1}^{P-1} M^k \right) (\mathbf{1} - M) = (\mathbf{1} - M^P),$$

it follows that if M has no eigenvalues equal to unity, then we can write:

$$\mathbf{a}_{\text{cm}}^{1* \cdots *P} = (\mathbb{1} - M)^{-1} (\mathbb{1} - M^P) \mathbf{a}_{\text{cm}}.$$

If all of the eigenvalues $|\lambda_i(M)| < 1$, then as $P \rightarrow \infty$ we have:

$$\mathbf{a}_{\text{cm}}^{1* \cdots *P \rightarrow \infty} \rightarrow (\mathbb{1} - M)^{-1} \mathbf{a}_{\text{cm}}.$$

In other words, for such pdfs on $SE(N)$, after an infinite number of convolutions, the translational mean will remain finite even though the translational mean of the original PDF is not zero. This does not happen with pdfs on \mathbb{R}^N .

The results of this section (which we believe to be new) are used in the following section in our new statistical inverse kinematics algorithm.

5. INVERSE KINEMATICS BY BREADTH-FIRST MATCHING OF THE MEAN AND DESIRED POSE

In this section we introduce a new method for the inverse kinematics of discretely actuated manipulators using the concept of the mean of a workspace density function. The only stored information is \mathbf{a}_{cm}^k and M_k for $k = 1, \dots, P$. From this, we can easily calculate the mean of the workspace density of any subsection of the manipulator.

Let $g_{\text{cm}}^{k* \cdots *P}$ denote the expected value of the workspace density function for the segment of the manipulator consisting of the distal $P - k + 1$ modules. Given that we want the end of the manipulator to reach $g_{\text{des}} \in G$, we start at the base of the manipulator and ask which state of segment 1 minimizes $D(g^{(1)} \circ g_{\text{cm}}^{2* \cdots *P}, g_{\text{des}})$. After searching through all K possible values of g_1 , and the optimal $g^{(1)} = g_1$ is fixed, we proceed up the manipulator one unit. That is, we next minimize $D(g^{(2)} \circ g_{\text{cm}}^{3* \cdots *P}, g_{\text{des}})$. Since g_1 is fixed, K values of $g^{(2)} = g_1 \circ g_2$ are searched until the value of g_2 that minimizes is found. This procedure is performed by sequentially minimizing $D(g^{(k)} \circ g_{\text{cm}}^{(k+1)* \cdots *P}, g_{\text{des}})$ for all $k \in \{1, 2, \dots, P - 1\}$. When we get to $k = P$ we seek g_P such that $D(g^{(P)}, g_{\text{des}})$ is minimized. This algorithm is described in the flowchart at the end of this section.

The motivation behind this algorithm is that by continually having the remaining freedom of the manipulator try to put the expected value of the remaining density over the desired position and orientation, the chances of hitting the goal are high. Of course since in this algorithm we do not keep the whole workspace density function but rather only one of its moments, we would expect to lose some of the nice features of the extended Ebert-Uphoff algorithm. One clear example of this is that if the workspace density function for any segment of the manipulator has crescent-shaped support, then its mean value will be at a location with zero density. This means that our method will necessarily fail for such manipulators. However, if the workspace density functions for all segments of the manipulator have their mean and mode (peak value) close to each other, we expect our method to work well. This is illustrated in the next section with numerical examples.

Algorithm Flowchart:

- (1) Start.
- (2) Input number of modules.
- (3) Find and store the rotation matrices, position vectors and homogeneous transform of the end effector for all possible configurations of one module.
- (4) Find and store the 'means' of rotation matrices and position vector from all possible configurations.
- (5) Input the desired positions.
- (6) Set a temporary homogeneous transform matrix for the end effector.
- (7) Begin loop of the new inverse kinematics algorithm with breath-first search, starting with an active module at the base of the manipulator.
- (8) Compare and find the best configuration of the active module (which gives the shortest distance between the end effector and the desired position), using the 'mean' stored from (4) with the breath-first search algorithm.
- (9) Store the selected configuration of that active module.
- (10) Move active module to the next module in the manipulator.
- (11) If the active module is the second to last module or the last one then go to (12), otherwise go back to (8).
- (12) Find and store the best possible configurations for the last two modules (or last module) which yields the shortest distance between the end effector and the desired position, using all possible configurations with the breath-first search algorithm.
- (13) Use the stored configurations of each module to efficiently calculate the forward kinematics for the manipulator.
- (14) End.

6. NUMERICAL RESULTS

In this section, we report the results of the numerical simulations performed to evaluate the algorithm presented in the previous section.

The accuracy in which a 10-link planar revolute manipulator with eight evenly-spaced states can reach a set of randomly generated target positions is illustrated in Fig. 2. The desired target position vectors under evaluation are:

$$\begin{pmatrix} 0.5000 \\ 0.5000 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} -0.5405 \\ 0.5915 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} -0.3163 \\ 0.7277 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} 0.2000 \\ -0.2000 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0.8121 \\ 0.1720 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} -0.8600 \\ -0.0500 \\ 0 \end{pmatrix}.$$

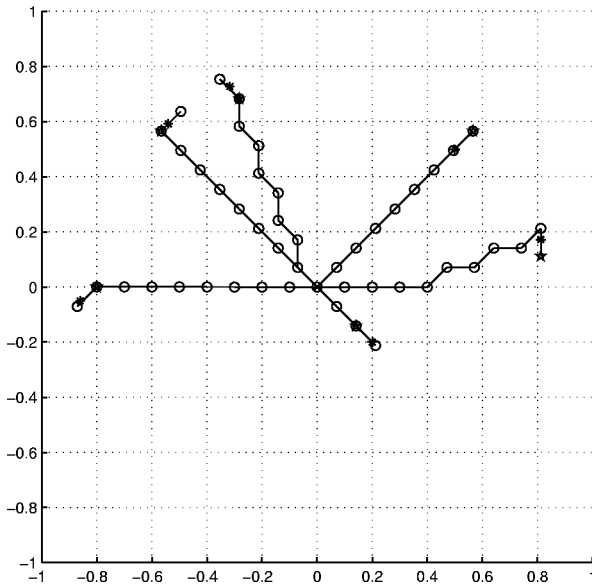


Figure 2. Demonstration of our method with a planar revolute manipulator.

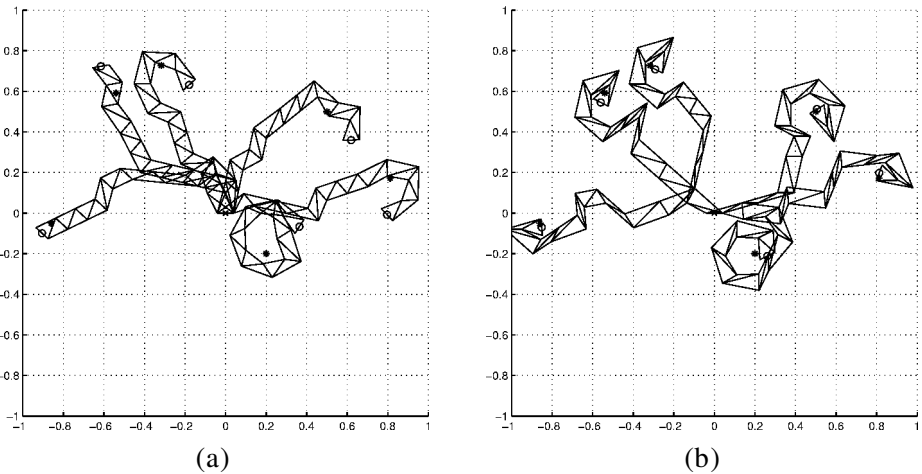


Figure 3. Demonstration of our method with a planar VGT manipulator.

Where the asterisks ‘*’ represent the desired positions, pentagons ‘★’ represent the end-effector positions and circles ‘o’ represent the joint positions. The base of the manipulator is located at the coordinates (0, 0).

The ability of a 10-module planar variable geometry truss (VGT) manipulator with two different sets of leg lengths to reach the same set of random target positions is shown in Fig. 3. Here the asterisks ‘*’ represent the desired positions and circles ‘o’ represent the end-effector positions. The base of the manipulator is located at the coordinates (0, 0).

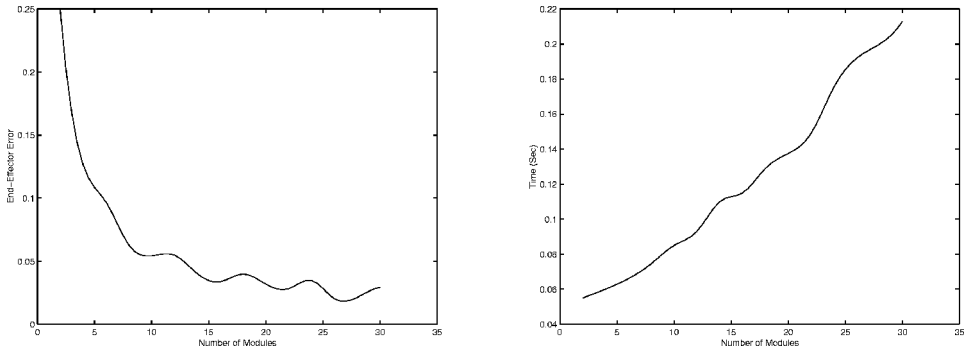


Figure 4. End-effector error and time to compute our method as a function of the number of modules (for manipulators of the kind in Fig. 2).

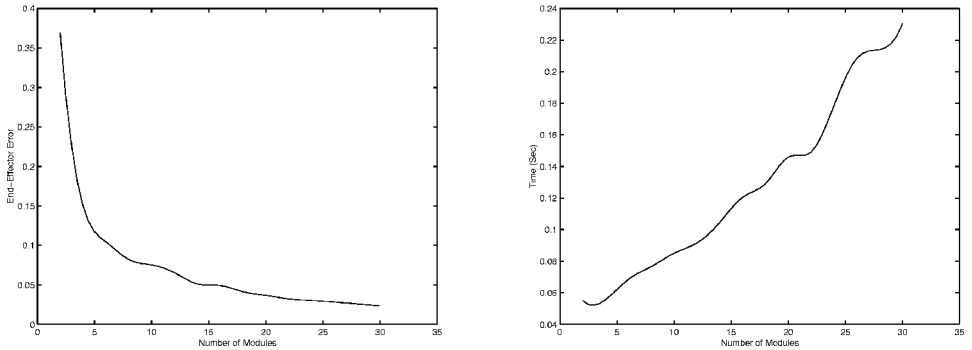


Figure 5. End-effector error and time to compute our method as a function of the number of modules (for manipulators of the kind in Fig. 3).

In Fig. 3a, the minimum and maximum manipulator lengths are $(l_{\min}, l_{\max}) = (1, 1.5)$. In Fig. 3b, the lengths are $(l_{\min}, l_{\max}) = (1, 1.7)$. The reason for the poor performance in Fig. 3a in contrast to the better performance in Fig. 3b is that the pdfs for the manipulator in Fig. 3a have crescent-shaped support, whereas the arm in Fig. 3b has the mean frame of its pdf in a heavily populated region of frames.

In Figs 4 and 5, we present the results of numerical studies where the number of joints (or modules) in the two manipulator architectures considered above are allowed to vary. As can be seen, for cases when the workspace density functions for all sections of the manipulator have means in heavily populated regions of $SE(2)$, the end-effector error (scaled by manipulator length) goes down with the number of modules. In all cases, the time grows linearly in the number of modules. This contrasts to the exponential growth in computation time that would be used for brute force evaluation of all possible configurations of the arm and distance evaluation.

In Table 1, the values of the end-effector errors and time consumed which vary by the leg length and the number of modules for a 10-module planar VGT manipulator

Table 1.

Numerical results of end-effector error and time in the VGT manipulator which vary by number of modules and min–max manipulator length

Number of modules	End-effector error			Time		
	1:1.5	1:1.6	1:1.7	1:1.5	1:1.6	1:1.7
2	0.58460	0.47410	0.36968	0.02500	0.04650	0.05500
4	0.23020	0.13770	0.14992	0.05750	0.05800	0.05550
6	0.15050	0.10320	0.10449	0.06600	0.06850	0.06850
8	0.15680	0.12070	0.08209	0.07950	0.07700	0.07650
10	0.13780	0.11230	0.07522	0.08500	0.08800	0.08500
12	0.12290	0.09830	0.06551	0.09600	0.09350	0.09150
14	0.13130	0.09030	0.05106	0.10700	0.10450	0.10450
16	0.12300	0.08740	0.04923	0.12350	0.11300	0.12050
18	0.10020	0.07340	0.04129	0.14550	0.12650	0.12900
20	0.09320	0.06730	0.03669	0.13700	0.14200	0.14600
22	0.07710	0.06300	0.03221	0.16750	0.15900	0.14900
24	0.06910	0.05840	0.03017	0.17850	0.17600	0.17850
26	0.06110	0.05200	0.02830	0.19500	0.19500	0.20850
28	0.06100	0.05240	0.02596	0.21150	0.21400	0.21400
30	0.05690	0.04630	0.02330	0.23900	0.22800	0.23050

are shown. The simulations are implemented in Matlab on an IBM compatible machine with an Intel Pentium III 733 MHz processor. Every value is an average value over 50 samples.

7. DESIGN

The algorithm presented in this paper for inverse kinematics is applicable both in 2D and 3D. In order to demonstrate inverse kinematics algorithms, we have built a full size three-module prototype of a new 3D manipulator using two-state pneumatic actuators. Furthermore, we have designed a new kind of discrete rotating joint controlled by three binary actuators. We briefly describe these designs below.

7.1. Design of a discrete rotating joint

We have designed a discrete rotating joint using three binary actuators to control the orientation and direction. We use a set of three gears. Two of these gears are mounted along with two rotary actuators on the base and the third is mounted on the module that is rotated by this actuator. One of two rotary actuators is used as a fluid-filled damper in order to decrease speed and reduce vibration. A series of holes aligned in a circular pattern on the module act as the position controller. While the rotary actuators and gear set are in operation, we use a compact linear binary

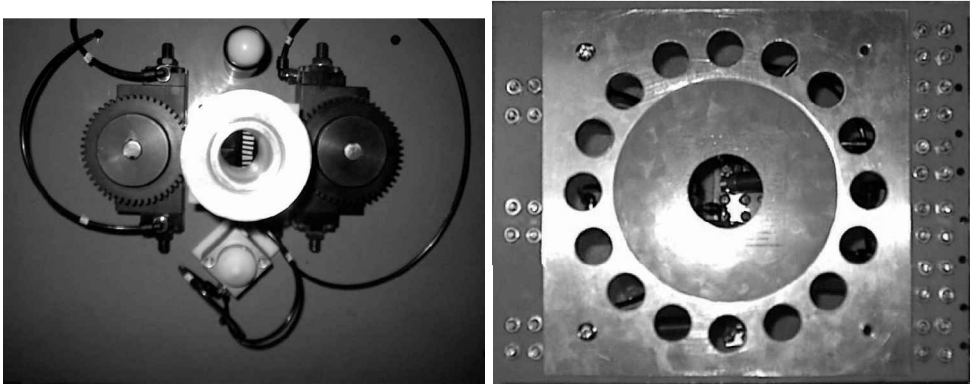


Figure 6. (Left) Rotary actuator comprised of a dashpot, linear actuator and gearset. (Right) A circular set of holes on the adjacent (rotated) module.

actuator located between the base and rotor as the stopper. A cone shape is attached to the tip of the actuator, which is inserted into each hole to stop the rotor. The base and rotor are shown in Fig. 6.

7.2. Design of a new spatial binary-actuated manipulator

This design is influenced by the advantages and disadvantages of previously built binary manipulators. The new design uses 3-bit planar binary VGT modules stacked on top of each other with a discretely actuated rotating joint between each module. Figure 7 illustrates some configurations of the new spatial binary-actuated manipulator. The prototype consists of four 3-bit binary VGT modules with discrete rotating joints between the modules. From a structural analysis of the manipulator, we found that the module which is the closest to the base should have the largest size. The following modules get smaller in size towards the end effector. We have built a full size prototype using two-state pneumatic actuators. PC interfacing with a relay switchboard triggers the solenoid valves and controls each actuator, and this prototype is lightweight as illustrated in Fig. 8. In order to reduce the vibration occurring due to the high speed of the pneumatic actuators, dashpots have to be introduced in parallel with the actuators in each module.

8. EXPERIMENTS AND RESULTS

A 3-bit planar VGT module has eight (2^3) possible reachable points. Therefore, the new discretely actuated manipulator can reach $2^{(3 \times N1)} \times (N2 \times N3)$ points. $N1$ is the number of 3-bit binary VGT modules, $N2$ is the number of rotating joints, and $N3$ is the number of states in each joint. We have built and tested the prototype manipulator. Although the load-bearing ability of the structure is a main concern due to the capacity of the actuators, overall the experiments with the manipulator were very successful.

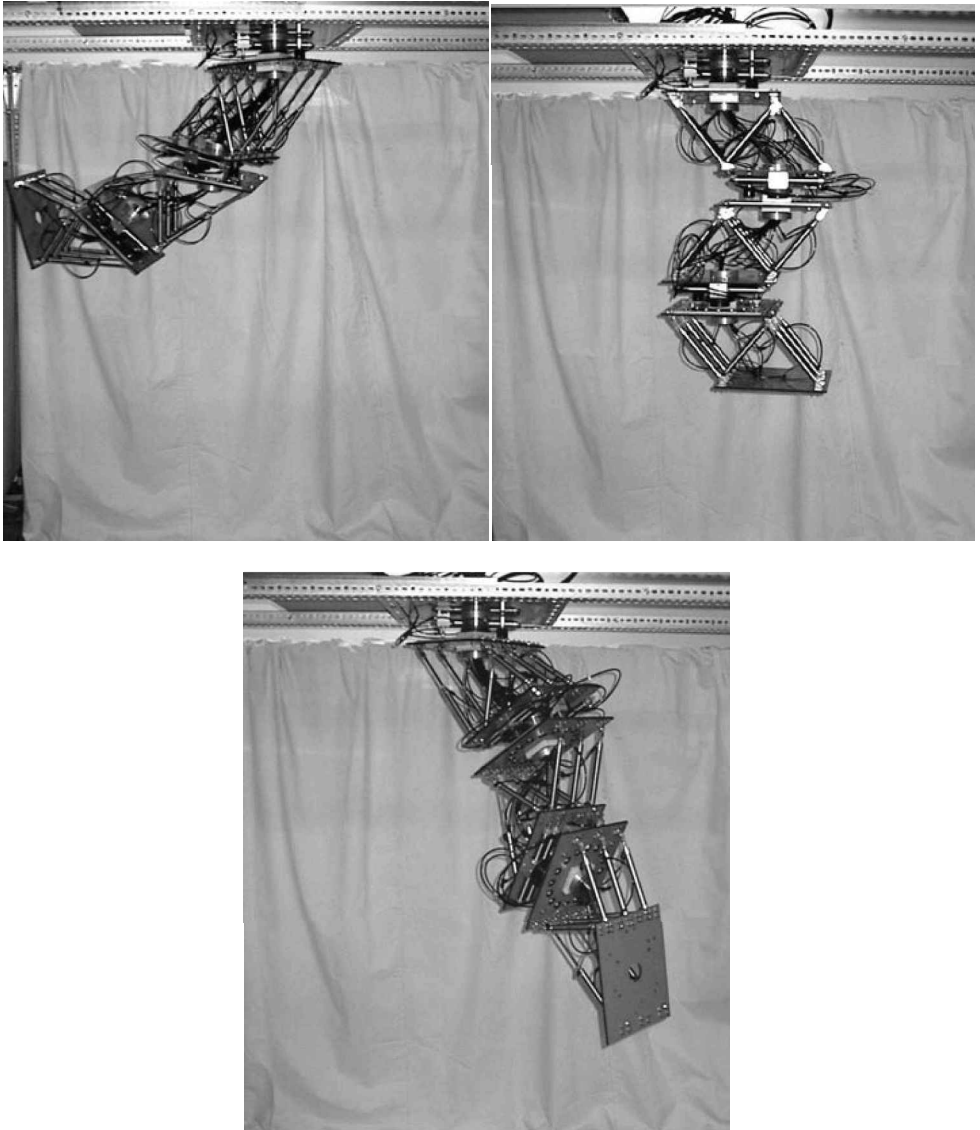


Figure 7. Configurations of the new spatial binary-actuated manipulator.

9. CONCLUSION

In this paper we presented a new technique for the inverse kinematics of highly actuated manipulators with discrete states. The method is based on the sequential minimization of a measure of distance between the mean of a workspace density function and the desired position of the end effector. Numerical results illustrate the usefulness of the method. We also describe a new design of a 3D discretely actuated robot manipulator powered by binary actuators. An advantage of this design over



Figure 8. The light weight of the new manipulator is illustrated in comparison with traditional manipulators.

traditional manipulators is its light weight. Our further experiments will involve implementation of the inverse kinematics algorithm presented in this paper with this new manipulator.

Acknowledgements

This work was supported by NSF grant RHA-97-31720 and a scholarship from the Royal Thai Government.

REFERENCES

1. J. M. McCarthy, *An Introduction to Theoretical Kinematics*. MIT Press, Cambridge, MA (1991).
2. R. M. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor, MI (1994).
3. J. Angeles, The design of isotropic manipulator architectures in the presence of redundancies, *Int. J. Robotics Res.* **11** (3), 196–201 (1992).
4. U. Basavaraj and J. Duffy, End-effector motion capabilities of serial manipulators, *Int. J. Robotics Res.* **12** (2), 132–145 (1993).
5. F. C. Park and R. W. Brockett, Kinematic dexterity of robotic mechanisms, *Int. J. Robotics Res.* **13** (1), 1–15 (1994).

6. T. Yoshikawa, Manipulability of robotic mechanisms, *Int. J. Robotics Res.* **4** (2), 3–9 (1985); also see *Foundations of Robotics: Analysis and Control*. MIT Press, Cambridge, MA (1990).
7. P. Diaconis, *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, Hayward, CA (1988).
8. D. L. Pieper, The kinematics of manipulators under computer control, PhD Dissertation, Stanford University (1968).
9. B. Roth, J. Rastegar and V. Scheinman, On the design of computer controlled manipulators, in: *Proc. 1st CISM-IFTMM Symp. on Theory and Practice of Robots and Manipulators*, pp. 93–113 (1973).
10. A. Koliskor, The 1-coordinate approach to the industrial robots design, in: *Information Control Problems in Manufacturing Technology: Proc. 5th IFAC/IFIP/IMACS/IFORS Conf.*, Suzdal, USSR, pp. 225–232 (preprint) (1986).
11. S. E. Salcudean, private communication.
12. A. Kumar and K. J. Waldron, Numerical plotting of surfaces of positioning accuracy of manipulators, *Mech. Mach. Theory* **16** (4), 361–368 (1980).
13. D. Sen and T. S. Mruthyunjaya, A discrete state perspective of manipulator workspaces, *Mech. Mach. Theory* **29** (4), 591–605 (1994).
14. G. S. Chirikjian, A binary paradigm for robotic manipulators, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, pp. 3063–3069 (1994).
15. G. S. Chirikjian, Kinematic synthesis of mechanisms and robotic manipulators with binary actuators, *ASME J. Mech. Design* **117**, 573–580 (1995).
16. I. Ebert-Uphoff, On the development of discretely-actuated hybrid-serial-parallel manipulators, PhD Dissertation, Johns Hopkins University (1997).
17. A. B. Kyatkin and G. S. Chirikjian, Synthesis of binary manipulators using the Fourier transform on the Euclidean group, *J. Mech. Design* **121**, 9–14 (1999).
18. I. Ebert-Uphoff and G. S. Chirikjian, Efficient workspace generation for binary manipulators with many actuators, *J. Robot. Syst.* **12** (6), 383–400 (1995).
19. G. S. Chirikjian and I. Ebert-Uphoff, Numerical convolution on the Euclidean group with applications to workspace generation, *IEEE Trans. Robotics Automat.* **14** (1), 123–136 (1998).
20. D. S. Lees and G. S. Chirikjian, An efficient trajectory planning method for binary manipulators, in: *ASME Mechanisms Conf.*, 96-DETC/MECH-1161 (1996) (9 pages on CD ROM).
21. D. S. Lees and G. S. Chirikjian, An efficient method for computing the forward kinematics of binary manipulators, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 1012–1017 (1996).
22. D. S. Lees and G. S. Chirikjian, A combinatorial approach to trajectory planning for binary manipulators, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 2749–2754 (1996).
23. G. S. Chirikjian and D. S. Lees, Inverse kinematics of binary manipulators with applications to service robotics, in: *Proc. IROS '95*, Pittsburgh, PA, Vol. 3, pp. 65–71 (1995).
24. G. S. Chirikjian, Inverse kinematics of binary manipulators using a continuum model, *J. Intel. Robot. Syst.* **19**, 5–22 (1997).
25. I. Ebert-Uphoff and G. S. Chirikjian, Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 139–145 (1996).
26. G. S. Chirikjian and S. Zhou, Metrics on motion and deformation of solid models, *J. Mech. Design* **120**, 252–261 (1998).
27. P. Fanghella and C. Galletti, Metric relations and displacement groups in mechanism and robot kinematic, *J. Mech. Design* **117**, 470–478 (1995).

28. K. Kazerounian and J. Rastegar, Object norms: a class of coordinate and metric independent norms for displacement, *Flexible Mechanisms, Dynam. Anal.* **47**, 271–275 (1992).
29. J.M.R. Martinez and J. Duffy, On the metrics of rigid body displacement for infinite and finite bodies, *ASME J. Mech. Design* **117**, 41–47 (1995).
30. M. efran, V. Kumar and C. Croke, Metrics and Connections for rigid-body kinematics, *Int. J. Robotics Res.* **18** (2), 243–258 (1999).
31. G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. CRC Press, Ann Arbor, MI (2000).
32. U. Grenander, M. I. Miller and A. Srivastava, Hilbert–Schmidt lower bounds for estimators on matrix Lie groups for ATR, *IEEE Trans. Pattern Anal. Machine Intel.* **20** (8), 790–802 (1998).
33. K. S. Arun, T. S. Huang and S. D. Blostein, Least-squares fitting of two 3-D point sets, *IEEE Trans. Pattern Anal. Machine Intel.* **9** (5), 698–700 (1987).
34. D. W. Eggert, A. Lorusso and R. B. Fischer, Estimating 3-D rigid body transformations: a comparison of four major algorithms, *Machine Vision Appl.* **9** (5-6), 272–290 (1997).
35. D. Goryn and S. Hein, On the estimation of rigid-body rotation from noisy data, *IEEE Trans. Pattern Anal. Machine Intel.* **17** (12), 1219–1220 (1995).
36. R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya and M. B. Kim, Pose estimation from corresponding point data, *IEEE Trans. Syst. Man Cybernet.* **19** (6), 1426–1446 (1989).
37. K. Kanatani, Analysis of 3-D rotation fitting, *IEEE Trans. Pattern Anal. Machine Intel.* **16** (5), 543–549 (1994).
38. S. H. Joseph, Optimal pose estimation in two and three dimensions, *Comp. Vision Image Understand.* **73** (2), 215–231 (1999).
39. A. Nádas, Least squares and maximum likelihood estimates of rigid motion, *IBM Res. Rep. RC 6945 (#29783)*, *Mathematics* (1978).
40. N. Ohta and K. Kanatani, Optimal estimation of three-dimensional rotation and reliability evaluation, *IEICE Trans. Information Syst.* **E81D** (11), 1247–1252 (1998).
41. S. H. Or, W. S. Luk, K. H. Wong and I. King, An efficient iterative pose estimation algorithm, *Image Vision Comput.* **16** (5), 353–362 (1998).
42. P. L. Rosin, Robust pose estimation, *IEEE Trans. Syst. Man Cybernet. B Cybernet.* **29** (2), 297–303 (1999).
43. S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. Pattern Anal. Machine Intel.* **13** (4), 376–380 (1991).
44. F. L. Markley, Attitude determination using vector observations and the singular value decomposition, *J. Astronaut. Sci.* **36** (3), 245–258 (1988).
45. V. L. Pisacane and R. C. Moore (Eds), *Fundamentals of Space Systems*. Oxford University Press, New York (1994).
46. M. D. Shuster and S. D. Oh, Three-axis attitude determination form vector observations, *J. Guidance Control Dynam.* **4** (1), 70–77 (1981).
47. G. Wahba, Problem 65-1, a least squares estimate of spacecraft attitude, *SIAM Rev.* **7** (3), 409 (1965).
48. G. Wahba *et al.*, Problem 65-1 (Solution), *SIAM Rev.* **8**, 384–386 (1966).
49. J. R. Wertz (Ed.), *Spacecraft Attitude Determination and Control*. Kluwer, Boston, MA (1978).

ABOUT THE AUTHORS

Jackrit Suthakorn was born in Thailand in 1973. He received a BE degree in Mechanical Engineering from Mahidol University, Bangkok Thailand in 1995. He received a MS degree in Mechanical Engineering from Michigan Technological University in 1998, and is presently pursuing a PhD degree in Mechanical Engineering at Johns Hopkins University. His research interests include binary robots and kinematics analysis of serial/parallel mechanisms.



Gregory S. Chirikjian was born in 1966 in New Brunswick, NJ. He received the BSE degree in Engineering Mechanics, the MSE degree in Mechanical Engineering and the BA degree in Mathematics, all from Johns Hopkins University, Baltimore, MD, in 1988. He then received the PhD degree from the California Institute of Technology, Pasadena, CA, in 1992. Since the summer of 1992, he has been with the Department of Mechanical Engineering, Johns Hopkins University, where he is now an Associate Professor. He was a 1993 NSF Young investigator, a 1994 Presidential Faculty Fellow and a 1996 recipient of the ASME Pi Tau Sigma

Gold Medal. His research interests include kinematic analysis, motion planning, design and implementation of 'hyper-redundant', 'metamorphic', and 'binary' manipulators. In recent years he has expanded the scope of his research to include applications of group theory in a variety of engineering disciplines.