Uniformly Distributed Elements of $SE(3)$ and their Application for Manipulator Design

MIYAHARA, Keizo
Department of Mechanical Engineering
Graduate School of Engineering
Osaka University
Suita, Osaka 565-0871, JAPAN
Email: miyahara@mech.eng.osaka-u.ac.jp

Gregory S. Chirikjian
Department of Mechanical Engineering
G.W.C.Whiting School of Engineering
Johns Hopkins University
Baltimore, Maryland 21218, USA
Email: gregc@jhu.edu

***Abstract*— This paper presents a computation method to generate a sequence of interpolated elements between arbitrary pair on the Special Euclidean group in three space, $SE(3)$. The necessity of this computation can be often found in robotic applications. The significance of the proposing method is that the uniformity of the sequential frame distribution on $SE(3)$ is guaranteed through a metric system defined for the rigid body motion.**

The computation method will be adapted to a kinematic synthesis problem of a class of robotic manipulators, "D-ARM" as an example application. A "Discretely Actuated Robotic Manipulator (D-ARM)", is any member of a class of robotic manipulators powered by actuators that have only discrete positional stable states such as solenoids. One of the most significant kinematic phenomena of D-ARMs is the discreteness of both input range and end-effector frames.

The conducted simulations demonstrate the feasibility of the synthesis procedure with the proposed frame computation method.

I. Metric System Defined for Rigid Body Motion

The Special Euclidean group of N dimensional space, $SE(N)$, is the semidirect product of $\mathbb{R}^N$ with the special orthogonal group, $SO(N)$. Elements of $SE(N)$ are denoted as $\mathbf{g} = (\mathbf{A}, \mathbf{a}) \in SE(N)$ where $\mathbf{A} \in SO(N)$ and $\mathbf{a} \in \mathbb{R}^N$. Any $\mathbf{g} \in SE(N)$ can be considered as a rigid-body motion, a pose, or a frame of reference in the N-dimensional Euclidean space, $\mathbb{R}^N$.

For any $\mathbf{g} = (\mathbf{A}, \mathbf{a})$ and $\mathbf{h} = (\mathbf{R}, \mathbf{r}) \in SE(N)$, the group law is written as $\mathbf{g} \circ \mathbf{h} = (\mathbf{AR}, \mathbf{a} + \mathbf{Ar})$ and $\mathbf{g}^{-1} = (\mathbf{A}^T, -\mathbf{A}^T\mathbf{a})$. Alternately, one may represent any element of $SE(N)$ as an $(N+1)\times(N+1)$ homogeneous transformation matrix of the form:

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

The action of a rigid-body motion $\mathbf{g} = (\mathbf{A}, \mathbf{a})$ on a vector $\mathbf{x} \in \mathbb{R}^N$ is defined by

$$\mathbf{g} \circ \mathbf{x} = \mathbf{Ax} + \mathbf{a}\,. \tag{2}$$

Park [1] has constructed a metric (measure of distance) between elements of $SE(3)$ as

$$d_P(\mathbf{g}_1, \mathbf{g}_2) = \left\| \log(\mathbf{H}(\mathbf{g}_1^{-1} \circ \mathbf{g}_2)) \right\|_W \tag{3}$$

where $\mathbf{g}_1, \mathbf{g}_2 \in SE(3)$, and

$$\left\|\mathbf{M}\right\|_W = \sqrt{\mathrm{tr}(\mathbf{MWM}^T)}$$

is the weighted matrix two-norm with

$$\mathbf{W} = \begin{pmatrix} \alpha\mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix},$$

and α is a user-specified length parameter that must be introduced to reconcile the difference between the units in translational and rotational quantities. This metric has left-invariant properties.

Another useful and easy-computable left-invariant metrics on the set of motions is defined by Chirikjian [2], [3] as following:

$$d^{(2)}(\mathbf{g}_1, \mathbf{g}_2) = \left\|\mathbf{g}_1 - \mathbf{g}_2\right\|_W\,, \tag{4}$$

and

$$\left\|\mathbf{g}_1 - \mathbf{g}_2\right\|_W^2 = 2\mathrm{tr}((\mathbf{I} - \mathbf{R}_1^T\mathbf{R}_2)\mathbf{W}_{3\times 3}) + w_{44}\left\|\bar{\mathbf{b_1}} - \bar{\mathbf{b_2}}\right\|^2, \tag{5}$$

where $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{W} = \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{0}^T & M \end{pmatrix}$, $M = \int_V \rho(\bar{\mathbf{x}})dV$ is the mass, $\mathbf{J} = \int_V \bar{\mathbf{x}}\bar{\mathbf{x}}^T\rho(\bar{\mathbf{x}})dV$, and $\rho(\bar{\mathbf{x}})$ be a real-valued non-negative function on $\mathbb{R}^3$ which satisfies the properties

$$0 < \int_{\mathbf{R}^3} \left\|\bar{\mathbf{x}}\right\|^m \rho(\bar{\mathbf{x}})dx_1dx_2dx_3 < \infty\,, \tag{6}$$

for any $m \geq 0$.

The properties of the metrics shown above are very convenient form for the computation purpose, and they are used extensively in the sequel to guarantee the uniform distribution of the frames.

II. Frame Interpolation Method

A computation method for a sequence of interpolated frames on $SE(3)$ will be proposed in this section. The dealing problem, which can be often found in robotics field, is stated as follows:

> Given two frames, say starting frame $\mathbf{H}_1$ and goal frame $\mathbf{H}_2$ ($\mathbf{H}_1, \mathbf{H}_2 \in SE(3)$), determine a

sequence of interpolated frames, $\mathbf{H}_{k(d)}$ for $d = 1, \ldots, n_{div}+1$, between the original two, so that every change between any adjacent two frames is small enough, and the interpolation is "even" with respect to some metric.

Here, n_{div} is the number of the interpolation, and "small enough" means that (25) holds. The generated sequence of frames will be used for the recursive computation of $\delta\mathbf{a}$ in Section III-E as an application.

A. Interpolation Method#1: Repetition of an identical rigid body motion

First, we examine an intuitive method of frame interpolation. This method is the basis of the next method, which is utilized in the simulation shown later, and might be more straightforward than the next one. Let $\mathbf{H}_1$ and $\mathbf{H}_2$ be the starting and the goal frames, respectively. We see that a rigid-body transformation from $\mathbf{H}_1$ to $\mathbf{H}_2$ is $\mathbf{H}_1^{-1}\mathbf{H}_2$. Let us denote $\mathbf{H}_1^{-1}\mathbf{H}_2$ as $\tilde{\mathbf{H}}$. Fig. 1 shows the geometric relationship between the transformations.

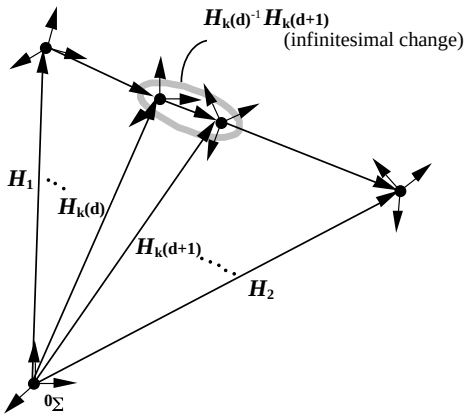


Fig. 1. Geometry of Frame Interpolation

We are going to divide this transformation, $\tilde{\mathbf{H}} \in SE(3)$, into n_{div} small transformations, $\delta\mathbf{H}$. We see that $\tilde{\mathbf{H}}$ can be written in the following form:

$$\tilde{\mathbf{H}} = \begin{pmatrix} \mathbf{R} & \mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{7}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{x} \in \mathbb{R}^3$. Moreover we know that $\mathbf{R}$ can be written in the following form with the Rodrigues' formula [4]:

$$\begin{aligned} \mathbf{R} &= \mathbf{e}^{\theta\hat{\boldsymbol{\Omega}}} \\ &= \mathbf{I} + \sin\theta\hat{\boldsymbol{\Omega}} + (1-\cos\theta)\hat{\boldsymbol{\Omega}}^2, \end{aligned} \tag{8}$$

where:

$$\theta = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R})-1}{2}\right)$$

$$\hat{\boldsymbol{\omega}} = vec\left(\hat{\boldsymbol{\Omega}}\right) = \begin{cases} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T & (\theta = 0) \\ \dfrac{1}{2\sin\theta}\boldsymbol{vec}(\mathbf{R}-\mathbf{R}^T) & (\text{otherwise}) \end{cases}.$$

Let $\mathbf{H}_{k(d)}$ be the sequence of the interpolated frames, for $d = 1, \ldots, n_{div}+1$. That is, there are n_{div} $\delta\mathbf{H}$s. Note that each of $\mathbf{H}_1$ and $\mathbf{H}_2$ is now $\mathbf{H}_{k(1)}$ and $\mathbf{H}_{k(n_{div}+1)}$, respectively. Without loss of generality, let us consider in the sequel the interpolated frames as the relative transformations with respect to $\mathbf{H}_1$. Namely:

$$\begin{aligned} \tilde{\mathbf{H}}_{k(1)} &= {\mathbf{H}_1}^{-1}\mathbf{H}_{k(1)} = \mathbf{I}_{4\times 4}; \\ \tilde{\mathbf{H}}_{k(d)} &= {\mathbf{H}_1}^{-1}\mathbf{H}_{k(d)}; \\ \tilde{\mathbf{H}}_{k(n_{div}+1)} &= {\mathbf{H}_1}^{-1}\mathbf{H}_{k(n_{div}+1)} = \tilde{\mathbf{H}}. \end{aligned} \tag{9}$$

Now we assume that $\delta\mathbf{H}$ is identical through the sequence, i.e.,

$$\delta\mathbf{H}^{n_{div}} = \tilde{\mathbf{H}}. \tag{10}$$

Similar to (7), we see that $\delta\mathbf{H}$ has the form:

$$\delta\mathbf{H} = \begin{pmatrix} \delta\mathbf{R} & \delta\mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{11}$$

where $\delta\mathbf{R} \in SO(3)$ and $\delta\mathbf{x} \in \mathbb{R}^3$. Therefore, we have:

$$\delta\mathbf{R} = \mathbf{e}^{\theta\hat{\boldsymbol{\Omega}}/n_{div}} \tag{12}$$

$$\delta\mathbf{x} = \left((\delta\mathbf{R})^{n_{div}-1} + \cdots + \delta\mathbf{R} + \mathbf{I}\right)^{-1}\mathbf{x}, \tag{13}$$

where the notation $(\cdot)^n$ means the matrix power in the contents of the parenthesis.

Using this, we can compute every $\tilde{\mathbf{H}}_{k(d)}$ as follows:

$$\begin{aligned} \tilde{\mathbf{H}}_{k(d)} &= (\delta\mathbf{H})^{d-1} \\ &= \begin{pmatrix} (\delta\mathbf{R})^{d-1} & \left((\delta\mathbf{R})^{d-2} + \cdots + \mathbf{I}\right)\delta\mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}, \end{aligned} \tag{14}$$

for $d = 1, \ldots, n_{div}+1$.

Fig. 2-a shows an example of frame interpolation by this method. In this example, we let:

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{I}, \\ \mathbf{H}_2 &= \begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\ n_{div} &= 1000. \end{aligned}$$

Graphics of the interpolated frames were output every 100 steps of the interpolation in the figure. A smooth interpolation was observed as shown in the figure. It is natural that repeated multiplication of a certain homogeneous transformation matrix forms a helical trajectory [5]. The rotational part of (14) expresses that it is the $d-1$-time recursion of the same motion, $\delta\mathbf{R}$. The translational part of the notation is pointing to the end point of the helix.

This helical transition means that the sequence does not travel "directly" from the initial frames to the desired frames in 3D space, particularly in position. In the next subsection, we consider an alternative method of interpolation, method #2, in which the sequence of frames travels along the shortest path.

B. Interpolation Method#2: Translation/Rotation decoupled motion

In the former method, we obtained a smooth interpolated sequence, but it formed a helix. This helical trajectory can not be avoided as long as we only use multiplication of a certain homogeneous transformation, $\delta\mathbf{H}$ as shown above. Here, we consider generating a sequence of interpolated frames, the origins of which lie on a line segment between the origins of the original two frames, $\mathbf{H}_1$ and $\mathbf{H}_2$.

The generating procedure is as follows: For orientation, we use the same interpolation as the former version. For positional part, however, we use a different computation in order to let the frames lie on a line segment. The form of the new $\tilde{\mathbf{H}}_{k(d)}$ is:

$$\begin{aligned}\tilde{\mathbf{H}}_{k(d)} &= \begin{pmatrix}\mathbf{R}_d & \mathbf{x}_d \\ \mathbf{0}^T & 1\end{pmatrix} \\ &= \begin{pmatrix}\delta\mathbf{R}^{d-1} & \frac{d-1}{n_{div}}\mathbf{x} \\ \mathbf{0}^T & 1\end{pmatrix}, \end{aligned} \tag{15}$$

for $d = 1, \ldots, n_{div}+1$. Note that $\mathbf{R}_d \in SO(3)$, $\mathbf{x}_d \in \mathbb{R}^3$, and thus $\tilde{\mathbf{H}}_{k(d)} \in SE(3)$. Fig. 2-b shows an example of frame interpolation with this new method. The given two frames are the same as the prior example. As shown in the figure, a smooth and direct interpolated trajectory was obtained with method #2. This interpolation method will be employed in the simulations shown later.

We confirm that the generated sequence of the frames is uniformly distributed with the proposed metric system. A distance in any metric is defined as a non-negative value. So we can compare the square of distances to guarantee the uniformity of the distribution. Applying this metric to two arbitrary adjacent frames, $\tilde{\mathbf{H}}_{k(d)}$ and $\tilde{\mathbf{H}}_{k(d+1)}$, in the interpolated sequence, we see that:

$$\begin{aligned}&d\left(\tilde{\mathbf{H}}_{k(d)}, \tilde{\mathbf{H}}_{k(d+1)}\right) = \\ &\sqrt{2\mathrm{tr}\left(\left(\mathbf{I}_{3\times 3} - \mathbf{R}_d^T\,\mathbf{R}_{d+1}\right)\mathbf{W}\right) + w_{44}\left\|\mathbf{x}_{d+1} - \mathbf{x}_d\right\|^2},\end{aligned} \tag{16}$$

for any d. We now see that the entities in (16) are constants for any d as follows:

$$\mathbf{R}_d^T\,\mathbf{R}_{d+1} = \delta\mathbf{R} \tag{17}$$

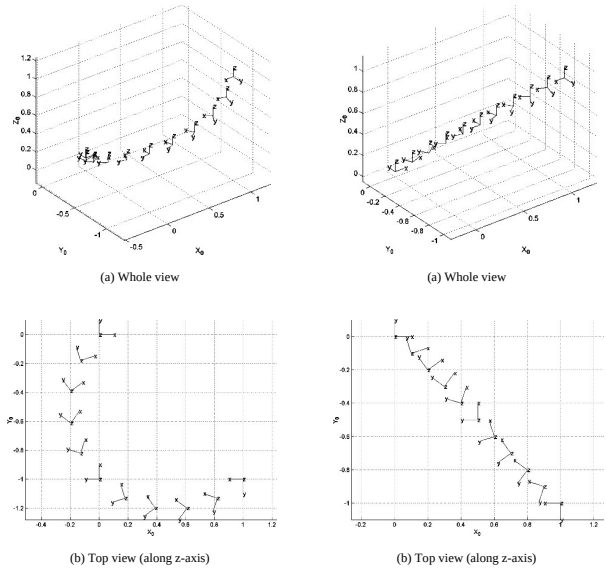
$$\left\|\mathbf{x}_{d+1} - \mathbf{x}_d\right\| = \frac{1}{n_{div}}\left\|\mathbf{x}\right\|. \tag{18}$$

Additionally, it can be shown that in method#1 the distance between any two adjacent frames is the same. The entities in the metric of method#1 are:

$$\mathbf{R}_d^T\,\mathbf{R}_{d+1} = \delta\mathbf{R} \tag{19}$$

$$\left\|\mathbf{x}_{d+1} - \mathbf{x}_d\right\| = \left\|\delta\mathbf{R}^{d-1}\,\delta\mathbf{x}\right\| = \left\|\delta\mathbf{x}\right\|, \tag{20}$$

for any d, and the last equality is a result of the fact that rotation matrices preserve the vector 2-norm.



(a) Whole view (a) Whole view

(b) Top view (along z-axis) (b) Top view (along z-axis)

(a) Method # 1 (b) Method # 2

Fig. 2. Result of Frame Interpolation

III. Example Application: Manipulator Design Problem

A. Definition of D-ARM with Actuator Categorization

For robotic manipulation, actuators are key components. Actuators can be recognized as belonging to one of the following two kinematic categories: the first one is continuously position controllable and accepts a continuous range of input command values. We denote this type of actuator as a "Continuous-Range-of-Motion Actuator" [6], or "Continuous Actuator." The other kind of actuator has only a finite number of discrete stable positions, and its input command range is discrete, as well. We denote this type of actuator as a "Discrete-Range-of-Motion Actuator", or "Discrete Actuator."

TABLE I
Actuator and Manipulator Categorization

Actuator	Example	Stable state = Input command	Manipulator
Continuous	Servomotor	Continuous range	C-ARM
Discrete	Solenoid, Pneumatic cylinder	Discrete range	D-ARM

Based on the actuator categorization shown above, we define a class of manipulators called the "Discretely Actuated Robotic Manipulator (D-ARM)" or "Discrete Arm" which is powered by discrete actuators. In particular, a "Binary Actuated Robotic Manipulator (B-ARM)" or "Binary Arm" is one with actuators that have only binary stable states. The categorization of D-ARM should be recognized as a generalization of the binary arm concept presented by Chirikjian [7].

Further, in contrast to D-ARM, let us call a manipulator a "Continuously Actuated Robotic Manipulator (C-ARM)", or "Continuous Arm", if the manipulator uses continuous actuators, as most standard robotic systems do.

B. Examples of D-ARM

One of the most fundamental examples is shown in Fig. 3. The 2D (two dimensional) B-ARM that has three bi-stable actuators is activated according to a three-digit binary number. The left (right) actuator is associated with the most (least) significant bit of the binary number, and the center actuator corresponds to the middle one. The binary bit "1" ("0") means full extension (contraction) of the actuator. By changing the binary number given to the controller, one of eight ($= 2^3$) possible configurations of the 3-bit B-ARM can be selected, and this means that one of eight possible frames of the end-effector can be reached with a certain binary number by the B-ARM.

The discussion above is also applicable for the 3D case in a similar manner. Fig. 4 shows a 3D B-ARM example. Total of 68.7 billion ($= 64^6$) end-frames are "reachable" by the B-ARM that consists of six stewart-type binary platforms. General s-state (multi-state) actuator case is also the scope of the discrete actuation paradigm as shown in [8].

One of the most significant kinematic phenomena of D-ARMs is the discreteness of both input range and end-effector frames as shown above.

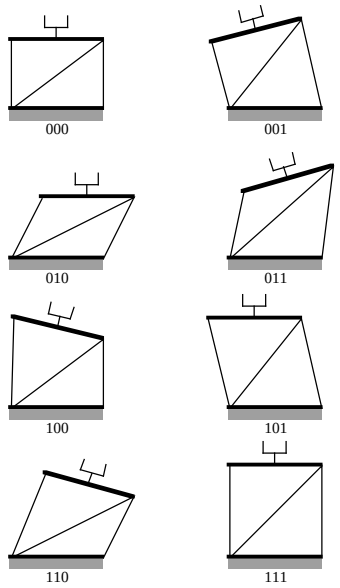


Fig. 3. An Example D-ARM (3-bit 2D Binary Parallel Platform)

C. Advantages and Applications of D-ARM

While discrete actuators are the key components of D-ARMs, they are also widely used as stand-alone motion sources in various applications. The variety of those applications originates from the following significant characteristics of discrete actuators:

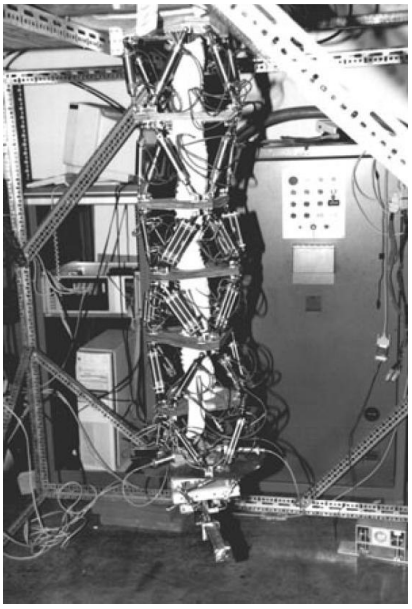


Fig. 4. An Example D-ARM (6 $\times$ 6-bit 3D Binary Parallel/Serial Hybrid Arm)

(1) Stability at each state without feedback loop
(2) High task repeatability
(3) Mechanism simplicity, including kinematic parameter adjustment (e.g., stroke stopper for a pneumatic cylinder)
(4) Minimal supporting devices, especially feedback systems
(5) Low cost and small volume due to (3) and (4)

Each state of a discrete actuator is mechanically stable because of simple internal mechanical constraints of parts, and the range of motion of a discrete actuator is also determined mechanically. The constraint is stable without an extensive, or even any, feedback loop (4). This results in the fundamental characteristics of discrete actuators: high stability (1) and high repeatability (2) which are mainly governed by the dimensional accuracy of the actuators parts. All of these characteristics of discrete actuators directly result in those of a D-ARM which is composed of discrete actuators.

Reconsidering continuous actuators from the point of view of cost efficiency, they seem to be "overkill" [9] when a task defines start/goal end-effector frames but the trajectory is less important as long as it is bounded. Conversely, discrete actuators could be a sufficient and cost effective solution for such a playback task if they have the ability to easily change kinematic parameters, such as stroke length for a cylinder. One of the most recent attempts to utilize the advantages of the discrete actuation can be found on the researches for robotic planetary explorers [10]. The mobile robot employs binary actuators to achieve a light weight and durable system with a simple and thus reliable controller, which are the same advantages of the discrete actuation system stated before.

D. kinetic synthesis problem

One of the most fundamental synthesis issues for manipulator design is to determine its kinematic parameters in order for reaching all the given desired frames.

In order to solve the design problem, several studies has been performed for B-ARM [7], [9], [11]. Such solutions, however, are intended to handle a positional kinematics only. The proposing kinematic synthesis process in this section deals with the desired end-effector orientations as well as the desired positions as follows:

> Given a D-ARM (base-line design) and finite sets of desired positions and orientations of the D-ARM's end-effector (desired frames: $\mathbf{H}_{ee} \in SE(N)$, where $N = 2(3)$ for 2D(3D) case), determine kinematic parameters (the vector $\mathbf{a}$) of the manipulator.

The design parameter $\mathbf{a}$ could be the link length for prismatic joints, or rotational angle for revolute joints of the manipulator. Note that this set of inverse kinematics problems becomes now on the Special Euclidean group, $SE(N)$, and that it is an extension of the previous B-ARM researches that discuss one on the Euclidean space, $\mathbb{R}^N$.

The flowchart of the proposing design process is shown in Fig. 5. The conceptual framework of the proposing synthesis method to solve the kinematic synthesis problem is based on an iterative computation with a numerically obtained Jacobian matrix of the given D-ARM and its weighted generalized inverse matrix.

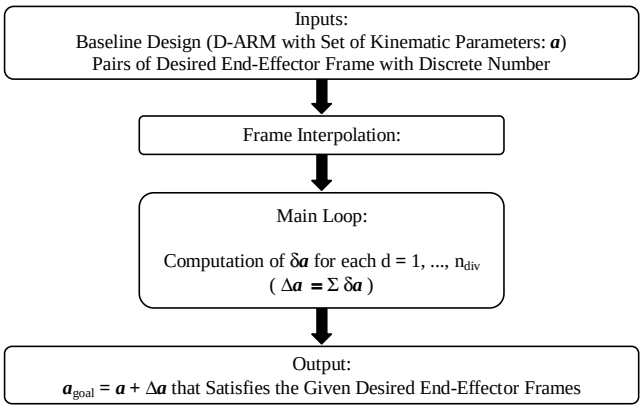


Fig. 5. Flowchart of the Proposing Kinematic Synthesis Method

In the iterative computation, the end-effector frame of a D-ARM is gradually shifted from its initial value to the given desired one, and at each step the kinematic parameters of the arm are updated in a computation that depends on the manipulator Jacobian matrix.

Now, a sequence of frames, $\mathbf{H}_{k(d)} \in SE(3)$ for $d = 1, \ldots, n_{div}+1$ between the initial end frame, $\mathbf{H}_{k(1)}$, and the desired frame, $\mathbf{H}_{k(n_{div}+1)}$, can be obtained with the method shown before, where n_{div} is the number of the fixed-steps in the interpolation. As discussed, the difference between any adjacent two frames, $\mathbf{H}_{k(d)}$ and $\mathbf{H}_{k(d+1)}$, can be small enough by choosing appropriate n_{div} for the frame sequences, i.e., the following condition holds:

$$\mathbf{H}_{k(d)}^{-1}\,\mathbf{H}_{k(d+1)} \approx \mathbf{I}\,, \tag{21}$$

for $d = 1, \ldots, n_{div}$, where $\mathbf{I}$ is the identity matrix.

The vector $\mathbf{a}$ represents a set of initial kinematic parameters. The vector $\mathbf{a}$ has the following structure for a general D-ARM with n_{st}-state actuators:

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{n_{st}} \end{pmatrix}, \tag{22}$$

where $\mathbf{a}_i \in \mathbb{R}^{n_{act}}$ is the set of kinematic parameters of the i^{th} state of the discrete actuators, and n_{act} denotes the number of actuators.

In the main loop, focusing on a certain pair of initial end frame and desired frame with an assigned discrete number, we calculate $\delta\mathbf{a}_{(d)}$ such that the end-effector frame changes from $\mathbf{H}_{k(d)}$ to $\mathbf{H}_{k(d+1)}$ for a certain d. In order for the renewal of $\mathbf{a}_{(d+1)}$, a method to compute $\delta\mathbf{a}_{(d)}$ from $\mathbf{H}_{k(d)}$, $\mathbf{H}_{k(d+1)}$, and $\mathbf{a}_{(d)}$ is required as following:

$$\mathbf{a}_{(d+1)} \leftarrow \mathbf{a}_{(d)} + \delta\mathbf{a}_{(d)}\,. \tag{23}$$

The summation of $\delta\mathbf{a}_{(d)}$ composes $\Delta\mathbf{a}$ in the flowchart. This iterative computation can be done associated with the forward kinematic function of the manipulator as shown below.

The forward kinematics for a D-ARM can be written in the form:

$$\mathbf{H}_{k(d)} = \mathbf{F}_k(\mathbf{a}_{(d)})\,. \tag{24}$$

For simplicity, the subscripts of $\mathbf{F}_k$ and $\mathbf{a}_{(d)}$ will be omitted in the sequel. Letting $n_{parm} = n_{st} \cdot n_{act}$, the small change from $\mathbf{H}_{k(d)}$ to $\mathbf{H}_{k(d+1)}$ is described as following:

$$\mathbf{H}_{k(d)}^{-1}\,\mathbf{H}_{k(d+1)} = \mathbf{I} + \sum_{i=1}^{n_{parm}} \mathbf{F}(\mathbf{a})^{-1}\frac{\partial \mathbf{F}}{\partial a_i}\delta a_i\,. \tag{25}$$

Letting $\delta\mathbf{x}_k$ be the small change through the sequence of interpolated frames for the k^{th} desired frame, together with (25), we have:

$$\begin{aligned} \delta\mathbf{x}_k &= \left(\mathbf{H}_{k(d)}^{-1}\,\mathbf{H}_{k(d+1)} - \mathbf{I}\right)^{\vee} \\ &\triangleq \mathbf{G}_k\delta\mathbf{a}\,, \end{aligned} \tag{26}$$

where $\mathbf{G}_k$ is the Jacobian matrix for the k^{th} desired frame.

The set of the obtained (26) for every k at a certain d, can be combined into one big equation as following:

$$\begin{aligned} \begin{pmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \vdots \\ \delta\mathbf{x}_{n_{frm}} \end{pmatrix} &= \begin{pmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \\ \vdots \\ \mathbf{G}_{n_{frm}} \end{pmatrix} \delta\mathbf{a} \\ &\triangleq \mathbf{G}\,\delta\mathbf{a}\,, \end{aligned} \tag{27}$$

where $\mathbf{G}$ is the "concatenated" Jacobian matrix of the given D-ARM at a certain d in the main loop. The rank of the system of

equations in (27) can be changed by the kinematic case of the original manipulator, i.e., sufficient, insufficient, or redundant.

The final result of the main loop in the flowchart in Fig. 5 is the following equation from (27):

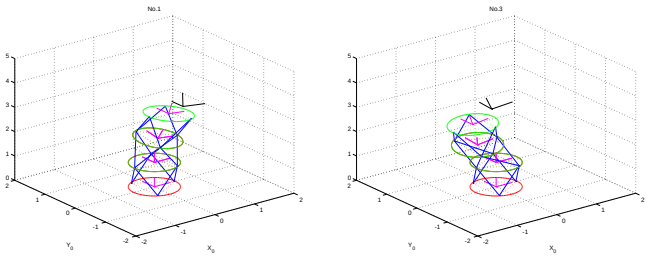
$$\delta \mathbf{a} \quad = \quad \mathbf{G}^{\dagger}\, \delta \mathbf{x}\,, \tag{28}$$

where † denotes the weighted generalized inverse matrix. According to the case of manipulator kinematics, the equation either has a unique, no, or an infinite number of exact solutions in general. No matter which kinematic case the system belongs to, we can solve the system of equations by utilizing the idea of a weighted generalized inverse matrix.

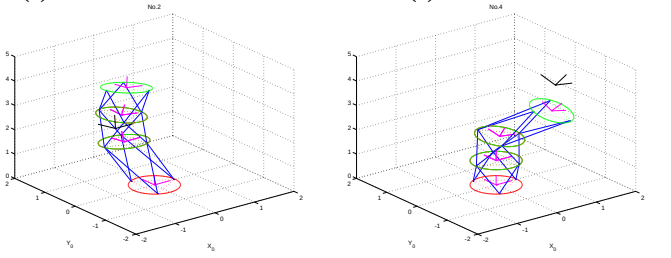
E. Simulation Results

Series of simulations are performed to demonstrate the feasibility of the algorithm discussed in the previous section. Through these simulations, it is found that the proposed synthesis algorithm is applicable to a wide variety of applications, and that the numerical errors were small enough and the computational time is short enough to design a real D-ARM even with a personal computer.

A typical example of the 3D sufficient case is shown in Fig. 6 and 7. Each link length of the manipulator is set as $1[m]$. The former figure shows the baseline design, and the latter one shows the final result of the kinematic synthesis process with 10^4 steps in the frame interpolation. The order of the position and the orientation error was less than $10^{-5}[m]$ and $10^{-5}[deg]$, respectively with 10^5 steps in the interpolation. The order of the computation time was $10^5[sec]$ with PC environment. These values of errors are also small enough, from the viewpoint of the dimensional accuracy of mechanical parts, to apply this synthesis method for actual D-ARM design even in the 3D case. According to the complexity analysis, the order of n_{div} dominates the computation time for ordinary D-ARM system.

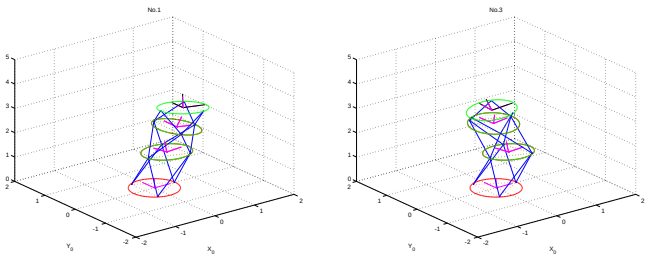


(a) Desired Frame # 1 (c) Desired Frame # 3

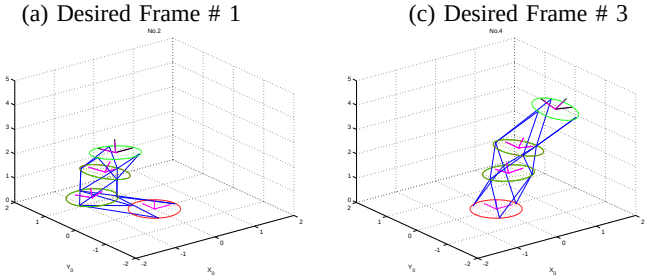


(b) Desired Frame # 2 (d) Desired Frame # 4

Fig. 6. Baseline Design Example (3-module 3D B-ARM, Suffi cient case)



(a) Desired Frame # 1 (c) Desired Frame # 3



(b) Desired Frame # 2 (d) Desired Frame # 4

Fig. 7. Synthesis Result Example (3-module 3D B-ARM, Suffi cient case)

REFERENCES

[1] F. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 117, pp. 48–54, Mar. 1995.

[2] G. Chirikjian and S. Zhou, "Metrics on motion and deformation of solid models," *Journal of Mechanical Design, Transaction of ASME*, vol. 120, pp. 252–261, 1998.

[3] G. Chirikjian and A. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton, FL: CRC Press, 2000.

[4] R. Murray, Z. Li, and S. Sastry, *A mathematical introduction to Robotic manipulation*. Boca Raton, FL: CRC press, 1993.

[5] O. Bottema and B. Roth, *Theoretical Kinematics*. New York, NY: Dover publication, 1979.

[6] I. Ebert-Uphoff, "On the development of Discretely-Actuated Hybrid-Serial-Parallel manipulators," Ph.D. Dissertation, Johns Hopkins University, Department of Mechanical Engineering, Baltimore, MD, May 1997.

[7] G. Chirikjian, "A binary paradigm for robotic manipulators," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. San Diego, CA: IEEE R&A, May 1994, pp. 3063–3069.

[8] K. MIYAHARA and G. S. Chirikjian, "D-arm: Precise and cost effective robotic manipulator," in *Proceedings of The 8th International Conference on Motion and Vibration Control (MOVIC 2006))*. Daejeon, Korea: KAIST, Aug. 2006, to be published.

[9] G. Chirikjian, "Kinematic synthesis of mechanisms and robotic manipulators with binary actuators," *Journal of Mechanical Design, Transaction of ASME*, vol. 117, pp. 573–580, Dec. 1995.

[10] V. Sujan and S. Dubowsky, "Design of a lightweight hyper-redundant deployable binary manipulator," *Journal of Mechanical Design, Transaction of ASME*, vol. 126, pp. 29–39, 2004.

[11] G. Chirikjian and I. Ebert-Uphoff, "Numerical convolution on the Euclidean group with applications to workspace generation," *IEEE Transaction on Robotics and Automation*, vol. 14, no. 1, pp. 123–136, Feb. 1998.