

A Jacobian-Based Algorithm for Planning Attitude Maneuvers Using Forward and Reverse Rotations

Sung K. Koh

Department of Mechanical Engineering,
Pohang University of Science and Technology,
Pohang, 790-784, Republic of Korea
e-mail: shkoh@postech.ac.kr

Gregory S. Chirikjian

Department of Mechanical Engineering,
Johns Hopkins University,
Baltimore, MD 21218
e-mail: gregc@jhu.edu

G. K. Ananthasuresh

Department of Mechanical Engineering,
Indian Institute of Science,
Bangalore, 560012, India
e-mail: suresh@mecheng.iisc.ernet.in

Algorithms for planning quasistatic attitude maneuvers based on the Jacobian of the forward kinematic mapping of fully-reversed (FR) sequences of rotations are proposed in this paper. An FR sequence of rotations is a series of finite rotations that consists of initial rotations about the axes of a body-fixed coordinate frame and subsequent rotations that undo these initial rotations. Unlike the Jacobian of conventional systems such as a robot manipulator, the Jacobian of the system manipulated through FR rotations is a null matrix at the identity, which leads to a total breakdown of the traditional Jacobian formulation. Therefore, the Jacobian algorithm is reformulated and implemented so as to synthesize an FR sequence for a desired rotational displacement. The Jacobian-based algorithm presented in this paper identifies particular six-rotation FR sequences that synthesize desired orientations. We developed the single-step and the multiple-step Jacobian methods to accomplish a given task using six-rotation FR sequences. The single-step Jacobian method identifies a specific FR sequence for a given desired orientation and the multiple-step Jacobian algorithm synthesizes physically feasible FR rotations on an optimal path. A comparison with existing algorithms verifies the fast convergence ability of the Jacobian-based algorithm. Unlike closed-form solutions to the inverse kinematics problem, the Jacobian-based algorithm determines the most efficient FR sequence that yields a desired rotational displacement through a simple and inexpensive numerical calculation. The procedure presented here is useful for those motion planning problems wherein the Jacobian is singular or null. [DOI: 10.1115/1.3007903]

Keywords: attitude control, fully-reversed sequences of rotations, noncommutativity of finite rigid rotations, Jacobian algorithm, inverse kinematics, motion planning

1 Introduction

A Jacobian-based algorithm to specify the orientation of a free rigid body undergoing fully-reversed (FR) sequences of rotations is presented in this paper. An FR sequence of rotations is defined as a series of rotations that consists of a series of initial forward rotations about the axes of a coordinate frame attached to a rigid body and a series of subsequent rotations that reverse the preceding forward rotations [1–4]. Since finite rotations of a rigid body are not commutative, finite FR rotations lead to a net orientation change of the rigid body.

The non-commutative property of finite rigid rotations is illustrated in Fig. 1. The cube in Fig. 1 undergoes successive rotations of 90 deg about the axes x , y , z , $-x$, $-y$, and $-z$. The orientation of the cube at the end of this fully-reversed operation is not the same as its initial orientation. We exploit six-rotation FR sequences of rotations to achieve a net orientation change of a rigid body because six-rotation FR sequence has sufficient degrees of freedom that are required to synthesize any possible orientations of the rigid body. Six-rotation FR sequences are composed of three initial forward rotations and three subsequent rotations in the reverse directions. The FR operation shown in Fig. 1 is an example of a six-rotation FR sequence. In this paper, we are concerned with

planning a correct FR sequence to re-orient the body to a desired orientation. The dynamics of the rigid body is beyond the scope of this paper.

The need for FR sequences of rotations stems from the working principle of the micro-scale device of Fig. 2(b) developed for quasistatic orientation control of a spacecraft. Li et al. [1] proposed a microscale *pseudo-momentum-wheel* as an alternative to the conventional torque generation mechanisms of spacecraft. The pseudo-momentum-wheel shown in Fig. 2(b) is an array of radially projecting microscale electro-thermal-compliant (ETC) actuators and generates a torque from the elastic deformation of its U-shape structure, as shown in Fig. 2(c). Upon application of a voltage to the ETC actuator, non-uniform temperature distribution on the actuator caused by Joule heating effects leads it to bend, as shown in Fig. 2(c). Therefore, when the pseudo-wheels are mounted on the surface of a rigid-cube, as shown in Fig. 2(a), the activation of the pseudo-wheel produces a torque to conserve the angular momentum of the system. However, when the actuators are deactivated quasistatically, they recover their original undeformed shape, which causes reverse directional rotations following the initial forward rotations. Therefore, the reverse rotations are inevitable when the rigid-cube is maneuvered by ETC pseudo-wheels. This provides the motivation for studying the kinematics of a rigid body that reorients its orientation through fully-reversed rotations according to the non-commutativity of finite rotations. According to the experiments with prototype pseudo-wheels [1], it takes about the order of milliseconds for the activation and deactivation of the ETC actuators. Therefore, we assume that the time required to turn on and off the ETC actuators is 1 ms throughout the examples presented in this paper. Since the moment of inertia

Contributed by the Design Engineering Division for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received June 18, 2007; final manuscript received May 20, 2008; published online December 12, 2008. Review conducted by Olivier A. Bauchau. Paper presented at the ASME 2007 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007), Las Vegas, NV, September 4–7, 2007.

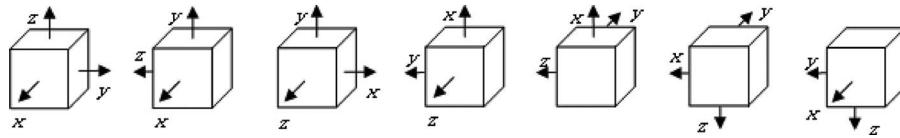


Fig. 1 Fully-reversed 90 deg rotations about the x -, y -, z -, $-x$ -, $-y$ -, and $-z$ -axes of a body-fixed coordinate frame

of pseudo-wheels is considerably smaller than that of the rigid-cube, the amount of input-rotations that can be obtained from the pseudo-wheels is limited in practice. However, we assume that the rigid-cube is capable of attaining any rotation about each axis of the body-attached coordinate frame in order to determine all FR sequences that synthesize a desired orientation. The realization of the FR motion determined in this manner is another issue because we need to consider the dynamics of the system that involves the moment of inertia of the system and the amount of torque that can be provided by physical torque generation mechanisms. Although we have studied the fully-reversed rotational motion to plan the motion of a miniature spacecraft, its kinematics is useful for the reorientation of any type of neutrally buoyant airborne and underwater vehicles maneuvered through fully-reversed rotations.

A large body of literature has been devoted to studying the orientation control of fully-actuated systems, and many robust and stable control algorithms have been developed [5–9]. In contrast to the conventional fully-actuated systems, the systems undergoing FR sequences of rotations have drawn attention since Li et al. [1] developed the pseudo-momentum-wheel. Using fully-reversed rotations for the reorientation of a rigid body, it is important to understand its forward kinematics and inverse kinematics in order to maneuver the rigid body with the particular motion. The forward kinematics of fully-reversed rotations was analyzed to explore the existence of solutions to the inverse kinematics problem by Koh et al. [3]. They also developed numerical and analytical techniques to solve the inverse kinematics problem [4]. While the solutions to a given inverse kinematics problem suggest a direct way of planning the orientation of a rigid body, this may not be a practical way to accomplish the given task if the magnitude of input-rotation is limited. In an attempt to develop efficient and practical motion planning algorithms, Koh et al. [2] proposed the pair-wise and single-sequence algorithms based on the leading order approximation of fully-reversed infinitesimal rotations. In this paper, we propose Jacobian-based algorithms that synthesize FR sequences for a given desired orientation through a simple numerical calculation.

Jacobian-based algorithms have been shown to be useful for planning the motion of the end effectors of robotic manipulators

[10] and for the determination of the orientation of spherical motors [11]. However, it is difficult to apply the Jacobian algorithm to a fully-reversed motion because the Jacobian can be singular in this case, as well as being a null matrix at the identity, which is usually taken as the initial configuration of a rigid body. In order to resolve this problem, we move the initial orientation from the identity to an arbitrary orientation using an FR sequence. The Jacobian at the chosen new initial orientation should not be singular. The path from the identity to the selected initial orientation is known because we selected the initial orientation using a specific FR sequence. Then, we need to determine the remaining path from the initial to final orientations. The remaining path is determined by applying the Jacobian algorithm from the selected initial orientation to the final orientation. The Jacobian algorithm implemented using six-rotation FR sequences attains the desired destination through either single step or multiple steps. It turned out that the Jacobian algorithm implemented using six-rotation FR rotations identifies exact solutions to the inverse kinematics problem. This feature of the Jacobian algorithm allows it to be faster for implementation in practice and more practical than existing algorithms such as the pair-wise and single-sequence algorithms.

The remainder of this paper is divided into several sections. Section 2 presents the background and the notation of FR sequences of rotations. Section 3 introduces a formulation of the Jacobian method using six-rotation FR rotations. Section 4 explores the performance of the single-step and multiple-step Jacobian methods. In Sec. 5, the performance of the Jacobian algorithm using six-rotation FR sequences is investigated in comparison with the pairwise and single-sequence algorithms. The contributions of the Jacobian algorithms are summarized in Sec. 6.

2 Definitions and Notation

A six-rotation fully-reversed rotation is mathematically defined as a mapping $FR_{abca-b-c^-}: \theta \in T^3 \rightarrow \mathbf{R}_{abca-b-c^-} \in SO(3)$, where

$$\mathbf{R}_{abca-b-c^-} = \mathbf{R}_a(\theta_a) \mathbf{R}_b(\theta_b) \mathbf{R}(\theta_c) \mathbf{R}_a(-\theta_a) \mathbf{R}_b(-\theta_b) \mathbf{R}_c(-\theta_c) \quad (1)$$

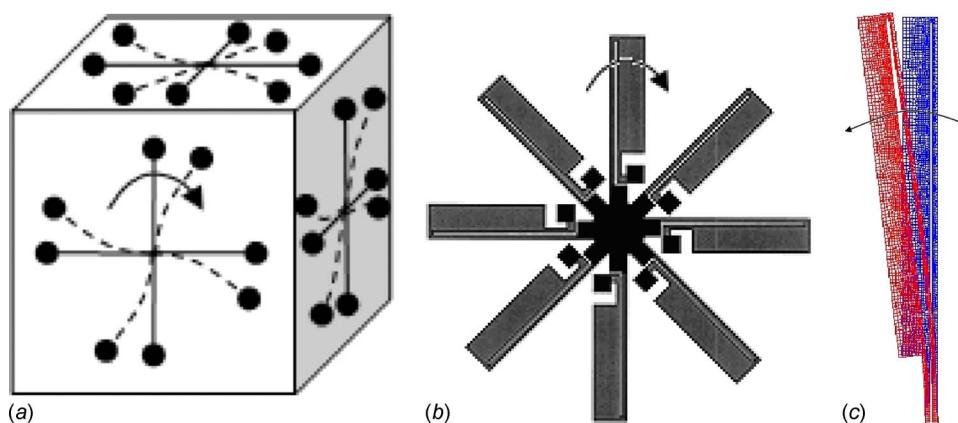


Fig. 2 Mechanisms for implementing FR rotations: (a) ETC pseudo-wheels attached to a rigid-body, (b) ETC pseudo-wheel, and (c) ETC actuator [2]

and T^3 is a product space of three copies of the circle (or one-torus), defined by the angles $\boldsymbol{\theta} = \{-\pi < \theta_a \leq \pi, -\pi < \theta_b \leq \pi, -\pi < \theta_c \leq \pi\}$; $SO(3) = \{\mathbf{R} \in \mathfrak{R}^{3 \times 3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = +1\}$ is the special orthogonal group whose elements describe orientations of a rigid body in a three-dimensional space. \mathbf{R}_a represents a rotation about the a -axis of the body-fixed coordinate frame by an angle θ_a and $\mathbf{I} \in SO(3)$ denotes the identity matrix. The variables a, b , and c can take any labels indicating the axis of the body-fixed coordinate frame such as x, y , and z . Note that the order of rotations of $\mathbf{R}_{abca^{-}b^{-}c^{-}}$ in Eq. (1) takes into account the fact that the rotations take place about the axes of the body-fixed coordinate frame but the overall resulting rotational displacement is expressed in a space-fixed coordinate frame.

According to the Euler theorem [10,12], any rotational displacement $\mathbf{R} \in SO(3)$ is equivalent to a rotation about the axis of rotation $\hat{\mathbf{w}} \in so(3)$ by the angle of rotation $\phi \in [0, 2\pi]$. $so(3)$ is a set of 3×3 skew-symmetric matrices. This equivalence between the rotational displacement $\mathbf{R} \in SO(3)$ and the rotation defined by ϕ and $\hat{\mathbf{w}}$ leads to a mapping between $\mathbf{R} \in SO(3)$ and $\phi\hat{\mathbf{w}} \in so(3)$, which parameterizes the entire orientations in $SO(3)$. The exponential mapping $\exp: \phi\hat{\mathbf{w}} \in so(3) \rightarrow \mathbf{R} \in SO(3)$ enables the rotational displacement \mathbf{R} to be determined for a given rotation about the axis $\hat{\mathbf{w}}$ by the angle ϕ , and hence $\phi\hat{\mathbf{w}} \in so(3)$ is the exponential coordinate established for $SO(3)$. As a result, any orientation in $SO(3)$ can be expressed by the matrix exponential of $\phi\hat{\mathbf{w}} \in so(3)$. On the other hand, the inverse of the exponential mapping determines the direction of rotation $\hat{\mathbf{w}}$ and the angle of rotation ϕ for a given rotational displacement $\mathbf{R} \in SO(3)$. The log mapping $\log: \mathbf{R} \in SO(3) \rightarrow \phi\hat{\mathbf{w}} \in so(3)$ allows us to identify the exponential coordinate $\phi\hat{\mathbf{w}}$ that yields the rotational displacement $\mathbf{R} \in SO(3)$. The rotation angle ϕ and the rotation axis $\hat{\mathbf{w}}$ for a given rotational displacement $\mathbf{R} \in SO(3)$ are determined as follows:

$$\phi = \arccos\left(\frac{\text{trace}(\mathbf{R}) - 1}{2}\right) \quad (2)$$

$$\hat{\mathbf{w}} = \frac{1}{2 \sin \phi} (\mathbf{R} - \mathbf{R}^T) \quad (3)$$

The vector form of the rotation axis $\hat{\mathbf{w}} \in so(3)$ is denoted by either $[\hat{\mathbf{w}}]^V$ or $\mathbf{w} \in \mathfrak{R}^3$, and may be represented as follows:

$$\hat{\mathbf{w}} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \Leftrightarrow [\hat{\mathbf{w}}]^V = \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad (4)$$

where the superscript V indicates the vector form of the skew-symmetric matrix.

3 The Jacobian Algorithm Using Six-Rotation FR Sequences

In our planning algorithm, which finds the FR sequence to rotate a body through a finite rotation, we will use an “artificial time” parameter to parameterize an artificial path connecting the current and desired orientations of the body. Then, we will obtain the desired angles in the FR sequence by letting them vary with this artificial time parameter. Hence, when we refer to angular velocity, time derivative, or any dynamical quantity, it is only in the sense of this artificial time construct.

The Jacobian of the forward kinematics mapping of six-rotation FR rotations associates the angular velocity $\dot{\boldsymbol{\omega}} \in so(3)$ of a rigid body with the rate of change of the angles $\boldsymbol{\theta} \in T^3$ [10,12]. In order to establish a physically meaningful relationship between $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\omega}}$ based on the Jacobian of the mapping $\mathbf{R}_{abca^{-}b^{-}c^{-}}$, we consider the identity

$$\mathbf{R}\mathbf{R}^T = \mathbf{I} \quad (5)$$

which signifies that \mathbf{R} is an element of $SO(3)$. The time derivative of Eq. (5) leads to

$$\dot{\mathbf{R}}\mathbf{R}^T + \mathbf{R}\dot{\mathbf{R}}^T = 0 \quad (6)$$

which implies that an instantaneous angular velocity $\dot{\boldsymbol{\omega}} = \dot{\mathbf{R}}\mathbf{R}^T \in so(3)$ in a spatial coordinate frame is represented as a skew-symmetric matrix. If the angle of rotation in a fully-reversed motion traverses along a path $\boldsymbol{\theta}(t) \in T^3$, the corresponding angular velocity $\dot{\boldsymbol{\omega}}$ can be written as

$$\dot{\boldsymbol{\omega}} = \nabla_{\boldsymbol{\theta}} \mathbf{R}\mathbf{R}^T \dot{\boldsymbol{\theta}} = \left[\frac{\partial \mathbf{R}}{\partial \theta_a} \mathbf{R}^T \dot{\theta}_a + \frac{\partial \mathbf{R}}{\partial \theta_b} \mathbf{R}^T \dot{\theta}_b + \frac{\partial \mathbf{R}}{\partial \theta_c} \mathbf{R}^T \dot{\theta}_c \right] \quad (7)$$

where $\boldsymbol{\theta} = \{\theta_a, \theta_b, \theta_c\}^T$ is the angle of rotation in FR sequences of rotations.

By defining the Jacobian $\mathbf{J}(\boldsymbol{\theta})$ as follows:

$$\mathbf{J} = \left[\left(\frac{\partial \mathbf{R}}{\partial \theta_a} \mathbf{R}^T \right)^V, \left(\frac{\partial \mathbf{R}}{\partial \theta_b} \mathbf{R}^T \right)^V, \left(\frac{\partial \mathbf{R}}{\partial \theta_c} \mathbf{R}^T \right)^V \right] \quad (8)$$

Equation (7) may be written in a vector form as follows:

$$\dot{\boldsymbol{\omega}}(t) = \mathbf{J}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (9)$$

where $\dot{\boldsymbol{\theta}} = \{\dot{\theta}_a, \dot{\theta}_b, \dot{\theta}_c\}^T$ is the speed of rotation and $\boldsymbol{\omega} \in \mathfrak{R}^3$ is the vector form of $\dot{\boldsymbol{\omega}} \in so(3)$. As shown in Eq. (9), the Jacobian $\mathbf{J}(\boldsymbol{\theta})$ defines a relationship between the speed of rotation $\dot{\boldsymbol{\theta}}$ in fully-reversed rotations and the instantaneous angular velocity $\boldsymbol{\omega}$. Therefore, Eq. (9) allows us to determine the angular velocity of the rigid body for a given speed of rotations. Note that the Jacobian \mathbf{J} is dependent upon the instantaneous configuration of the rigid body, which is decided by the angle of rotation $\boldsymbol{\theta} \in T^3$. If \mathbf{J} is invertible, the following ordinary differential equation is constructed and the angle of rotation $\boldsymbol{\theta}$ corresponding to the angular velocity $\boldsymbol{\omega}$ can be determined without solving the inverse kinematics problem:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}(\boldsymbol{\theta})^{-1} \boldsymbol{\omega}(t) \quad (10)$$

Once $\boldsymbol{\omega}(t)$ that defines a path of the rigid-body orientation to a desired orientation is established, the angle of rotation $\boldsymbol{\theta}_f$ that achieves a desired orientation is decided by solving the ordinary differential equation in Eq. (10). In this study, we solve Eq. (10) in an artificial time frame $t \in [0, 1] \subset \mathfrak{R}$ in order to determine an FR sequence that achieves a desired rotational displacement \mathbf{R}_f . Note that Eq. (10) determines an FR sequence where its resulting rotational displacement can be attained by rotating the rigid body at the angular velocity $\boldsymbol{\omega}(t)$ for a unit time period. Once the FR sequence for \mathbf{R}_f is determined, a physical time during which the FR sequence is executed can be estimated accounting for the dynamics of the system.

Let us define a Riemannian metric for $SO(3)$ by $\langle \dot{\mathbf{R}}\dot{\mathbf{R}} \rangle_{\mathbf{R}} = \boldsymbol{\omega}^T \boldsymbol{\omega}$, where $\boldsymbol{\omega} \in \mathfrak{R}^3$ and consider a curve $\mathbf{R}(t)$ between the initial orientation $\mathbf{R}_i = \mathbf{I}$ and the final orientation \mathbf{R}_f . Defining the length of the curve as follows:

$$l = \int_0^1 \langle \dot{\mathbf{R}}(t)\dot{\mathbf{R}}(t) \rangle^{1/2} dt = \int_0^1 (\boldsymbol{\omega}^T \boldsymbol{\omega})^{1/2} dt \quad (11)$$

an optimal path, *geodesic*, between \mathbf{R}_i and \mathbf{R}_f refers to a curve with minimum length. This optimal path follows a rotation indicated by $\phi\hat{\mathbf{w}} = \log[\mathbf{R}_f]$, where the direction of rotation $\hat{\mathbf{w}}$ and the rotation angle ϕ are determined by Eqs. (2) and (3). In order for a rigid body to achieve a rotational displacement indicated by $\phi\hat{\mathbf{w}}$, Eq. (10) is solved for a constant angular velocity $\boldsymbol{\omega}_{\text{opt}} = [\phi\hat{\mathbf{w}}]^V = \phi\mathbf{w} \in \mathfrak{R}^3$. Since an FR rotational operation is accomplished over a unit time period in the artificial time frame, $\phi\mathbf{w}$ is equivalent to the angular velocity of the rigid body, $\boldsymbol{\omega}_{\text{opt}}$, in this case. Then, Eq.

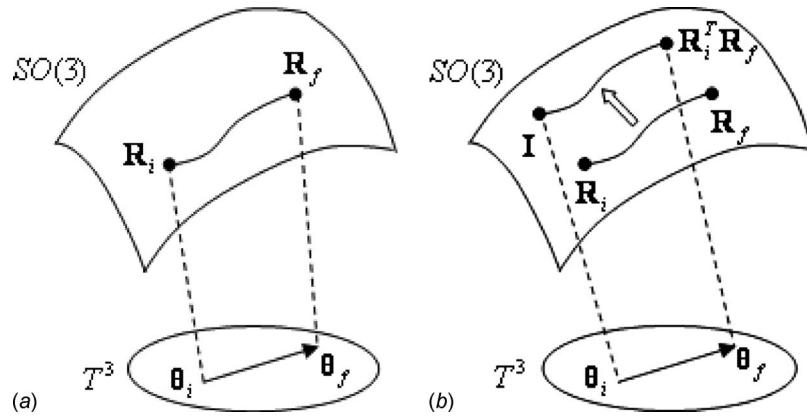


Fig. 3 Translation of the configuration of a rigid body in $SO(3)$. The initial orientation \mathbf{R}_i and the final orientation \mathbf{R}_f are translated to \mathbf{I} and $\mathbf{R}_i^T \mathbf{R}_f$, respectively, so that the rotation angle ϕ and rotation axis $\hat{\mathbf{w}}$ of $\log[\mathbf{R}_i^T \mathbf{R}_f]$ can be evaluated by Eqs. (2) and (3).

(10) determines $\boldsymbol{\theta}$ such that the rigid body achieves a rotational displacement that can be attained by the rotation $\exp[\phi\hat{\mathbf{w}}]$ performed over a unit time period about the rotation axis $\hat{\mathbf{w}}$ by the rotation angle ϕ in Eqs. (2) and (3). Therefore, it is assured that the final rotation angle at $t=1$ determined by Eq. (10) synthesizes the FR rotation that yields the rotational displacement $\exp[\phi\hat{\mathbf{w}}]$. Note that a real path constructed by the FR motion determined by Eq. (10) is not the same as the optimal path defined by $\exp[\phi\hat{\mathbf{w}}]$. The final orientation achieved at the end of the FR rotational operation is identical with the rotational displacement $\exp[\phi\hat{\mathbf{w}}]$.

A difficulty in applying the conventional Jacobian algorithm to fully-reversed rotations arises from the fact that the Jacobian of the forward kinematics mapping $FR_{abc-a-b-c}$ is singular and a null matrix at the identity on $SO(3)$. The singularity of $\mathbf{J}_{abc-a-b-c}$ at the identity can be understood by exploring the rotational displacement of a rigid body for a given infinitesimal $\Delta\boldsymbol{\theta} \in T^3$. When rotations are infinitesimally small, elements of rotation matrices can be approximated with the relationships $\sin(\Delta\boldsymbol{\theta}) \approx \Delta\boldsymbol{\theta}$ and $\cos(\Delta\boldsymbol{\theta}) \approx 1$. Then, the rotational displacement $\Delta\mathbf{R}_{abc} = \mathbf{R}_a(\Delta\theta_a)\mathbf{R}_b(\Delta\theta_b)\mathbf{R}_c(\Delta\theta_c)$ following the Euler parameterization [10,12] is approximated as

$$\Delta\mathbf{R}_{abc} \approx \begin{bmatrix} 1 & -\Delta\theta_c & \Delta\theta_b \\ \Delta\theta_c & 1 & -\Delta\theta_a \\ -\Delta\theta_b & \Delta\theta_a & 1 \end{bmatrix} \quad (12)$$

Similarly, the infinitesimal approximation of the FR rotation $FR_{abc-a-b-c}$ is represented as

$$\Delta\mathbf{R}_{abc-a-b-c} \approx \begin{bmatrix} 1 & \Delta\theta_a\Delta\theta_b & \Delta\theta_a\Delta\theta_c \\ -\Delta\theta_a\Delta\theta_b & 1 & -\Delta\theta_b\Delta\theta_c \\ -\Delta\theta_a\Delta\theta_c & \Delta\theta_b\Delta\theta_c & 1 \end{bmatrix} \quad (13)$$

As shown in Eq. (12), an infinitesimal rotational displacement by the Euler parameterization is dominated by first order terms, whereas infinitesimal FR rotation is governed by second order terms. This can be explained by a commutative property of infinitesimal rotations. Unlike finite rotations, infinitesimal rotations of a rigid body are commutative, and the first order terms in Eq. (12) are nullified by the first order effect of the subsequent rotations in Eq. (13). Therefore, FR sequences of infinitesimal rotations are approximated by the leading second order terms in Eq. (13). On the other hand, the tangent plane to $\mathbf{R} \in SO(3)$ at $\boldsymbol{\theta}=0$ is defined as

$$\mathbf{R}(\Delta\boldsymbol{\theta}) \approx \mathbf{I} + \mathbf{J} \cdot \Delta\boldsymbol{\theta} \quad (14)$$

In the case of FR rotations, as shown in Eq. (13), $\Delta\mathbf{R}_{abc-a-b-c} = \mathbf{R}_{abc-a-b-c}(\Delta\boldsymbol{\theta}) - \mathbf{I}$ contains only second order terms, which explains why $\mathbf{J}_{abc-a-b-c}$ is a null matrix at the identity. Equation (14) implies that \mathbf{J} indicates the rate of change of $\mathbf{R} \in SO(3)$ up to first order for a given infinitesimal $\Delta\boldsymbol{\theta} \in T^3$. Since $\mathbf{J}_{abc-a-b-c} = \mathbf{0}$ due to the commutativity of infinitesimal rotations, there is no rotational displacement by the first order effect, which is decided by the rate of change of $\mathbf{R}_{abc-a-b-c}$ at the identity.

In describing a rigid-body motion, the identity is usually taken as an initial orientation. However, as the Jacobian of $FR_{abc-a-b-c}$ is singular at the identity, the Jacobian equation in Eq. (9) does not hold at the identity, which makes the conventional Jacobian algorithm break down. Here, we suggest that any arbitrary orientation \mathbf{R}_i at which the Jacobian is not singular and the angle of rotation $\boldsymbol{\theta}_i$ corresponding to \mathbf{R}_i is already known can be selected as an initial orientation. Note that we determine \mathbf{R}_i using a particular FR sequence. Therefore, once \mathbf{R}_i is chosen using the FR sequence, we are aware how the orientation of the rigid body changes from the identity to the selected initial orientation \mathbf{R}_i . Then, the angle of rotation $\boldsymbol{\theta}_f$ corresponding to \mathbf{R}_f is determined by integrating Eq. (10) from the selected initial orientation $\mathbf{R}_i(\boldsymbol{\theta}_i)$. The final angle of rotation $\boldsymbol{\theta}_f$ attained in this manner enables adjustment of orientation from the identity to the desired orientation \mathbf{R}_f . Therefore, the desired angular velocity $\boldsymbol{\omega}(t)$ in Eq. (9) should indicate the orientation change from \mathbf{R}_i to \mathbf{R}_f . We obtain the desired angular velocity $\boldsymbol{\omega}(t)$ from the exponential coordinate of the rotational displacement from \mathbf{R}_i to \mathbf{R}_f , and it is maintained constant while integrating Eq. (10) over the time period $t \in [0, 1] \subset \mathcal{R}$.

The evaluation of the quantities defining the rotational displacement from \mathbf{R}_i to \mathbf{R}_f requires extra numerical computation. Therefore, we translate \mathbf{R}_i and \mathbf{R}_f to \mathbf{I} and $\mathbf{R}_i^T \mathbf{R}_f$, respectively, as shown in Fig. 3. This translation facilitates the evaluation of the rotation angle ϕ and the rotation axis $\hat{\mathbf{w}}$ of $\log[\mathbf{R}_i^T \mathbf{R}_f]$ for which a formulation is already well established in Eqs. (2) and (3). The Jacobian equation translated by \mathbf{R}_i^T is

$$\mathbf{R}_i^T \boldsymbol{\omega}(t) = \mathbf{R}_i^T \mathbf{J}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (15)$$

Therefore, in solving Eq. (15), although the angle of rotation varies from $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_f$, the rotation matrix describing the orientation of the rigid body traverses from the identity \mathbf{I} to the translated desired orientation $\mathbf{R}_i^T \mathbf{R}_f$. The performance of the proposed Jacobian

algorithm is verified through a number of numerical simulations that will be presented in the next section.

4 Performance of the Jacobian Algorithm Using Six-Rotation FR Rotations

Two algorithms based on the Jacobian equations in Eqs. (10) and (15) are presented in this section. The angle of rotation θ that yields a desired rotational displacement through FR rotations can be determined from Eq. (15). This algorithm is referred to as the “single-step Jacobian method” because the FR sequence obtained by solving Eq. (15) reaches the desired orientation in a single six-rotation FR sequence. According to the results of several numerical computations, the numerical solutions of the single-step Jacobian method converge to the analytical solutions developed for the synthesis of FR rotations [4]. On the other hand, the desired orientation can also be reached through an incremental path constructed in an optimal manner. Using this approach, the angle of rotation at each step is determined by the single-step Jacobian algorithm, and an incremental desired orientation is updated at each step such that the converged orientation at the current step is utilized as an initial orientation for the next step. This algorithm is referred to as the “multiple-step Jacobian method.”

4.1 Single-Step Jacobian Method. The performance of the single-step Jacobian method is investigated through numerical simulations in this section. Among 24 possible non-trivial six-rotation FR rotations, we consider the FR sequence $\mathbf{R}_{y\bar{x}\bar{z}-y\bar{x}} = \mathbf{R}_y(\theta_y)\mathbf{R}_z(\theta_z)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)\mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)$ as Koh et al. [3] has already verified that any feasible orientations of a rigid body can be attained through the FR operation $\mathbf{R}_{y\bar{x}\bar{z}-y\bar{x}}$. The formulation of the Jacobian-based algorithm addressed in Sec. 3 is implemented, as shown in the flow chart in Fig. 4. The variables shown in the flow chart in Fig. 4 are consistent with those in Sec. 3. The single-step Jacobian method is implemented in MATLAB [13] as it contains a built-in solver for an ordinary differential equation such as the Jacobian equation in Eqs. (10) and (15). As indicated in Eq. (15), it is convenient to evaluate the rotation angle ϕ and rotation axis $\hat{\mathbf{w}}$ of the angular velocity $\hat{\boldsymbol{\omega}}$ when the current orientation is translated by \mathbf{R}_i^T . Hence, the Jacobian equation in Eq. (15) is solved for a desired orientation $\mathbf{R}_i^T \mathbf{R}_f$ in Fig. 4. Note that the analytic form of $\mathbf{R}_{y\bar{x}\bar{z}-y\bar{x}}^T \dot{\mathbf{R}}_{y\bar{x}\bar{z}-y\bar{x}}$ is used for the evaluation of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ in Eqs. (10) and (15). If $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ becomes almost singular, then the pseudoinverse of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ is used to construct the Jacobian equation in Eq. (15). The singularity of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ is estimated by a condition number. As shown in Fig. 4, if the condition number of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ indicates that $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ is almost singular, the Jacobian equation $\dot{\boldsymbol{\theta}} = \mathbf{J}_{\text{pseudo}}^T \hat{\boldsymbol{\omega}}$ is solved instead of Eq. (15). $\mathbf{J}_{\text{pseudo}}^T$ denotes the pseudoinverse of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ and $\mathbf{J}_{\text{trans}}^T$ represents $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ translated by \mathbf{R}_i^T .

Not all 24 six-rotation FR rotations can synthesize all possible orientations in $SO(3)$. According to the analysis of Koh et al. [3], some FR rotations are capable of synthesizing only certain portions of orientations in $SO(3)$. If the orientation of a rigid body is adjusted through such an FR rotation, some orientations in $SO(3)$ are infeasible, and a boundary between feasible and infeasible orientations is formed. It was proven by Koh et al. [3] that the Jacobian of such FR rotations is singular on the boundary. Therefore, if a desired orientation is located in the infeasible region, although the orientation of the rigid body is initially driven in the feasible region, it eventually crosses the boundary to reach the desired orientation in the infeasible region. Therefore, the Jacobian algorithm fails on the boundary as the Jacobian equation in Eq. (10) can be established only when \mathbf{J} is invertible. In order to explore how the Jacobian algorithm performs in this case, we implemented it using the FR rotation $\mathbf{R}_{y\bar{x}\bar{z}-z\bar{z}}$, which covers only

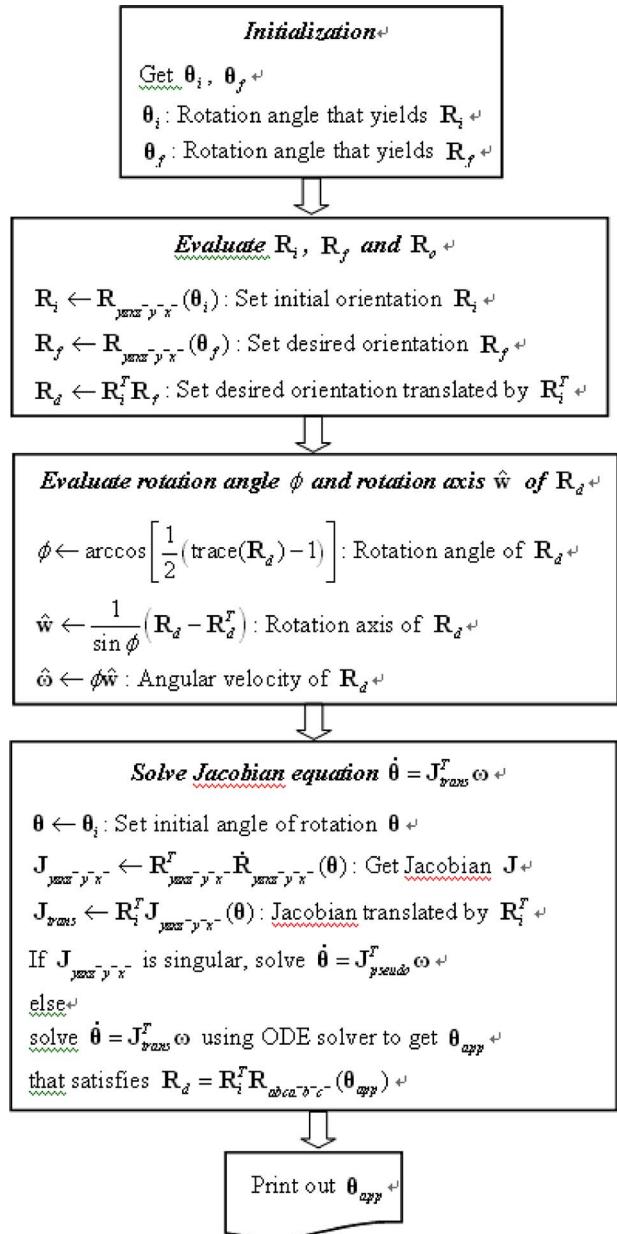


Fig. 4 Flow chart of the single-step Jacobian-based algorithm

certain portions of orientations in $SO(3)$. The existence of singularities of Jacobian is verified by examining the determinant of $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$, which is represented as

$$\det[\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}] = 2 \sin \theta_y (\cos \theta_x \sin \theta_y - \cos \theta_z \sin \theta_y - \cos \theta_y \sin \theta_x \sin \theta_z) \quad (16)$$

Therefore, the set of angles where the determinant in Eq. (16) is zero, and hence where $\mathbf{J}_{y\bar{x}\bar{z}-y\bar{x}}$ is singular, is defined when any of the following conditions hold:

$$\theta_y = 0, \quad \theta_y = \pi, \quad \theta_y = \tan^{-1}\left(\frac{\sin \theta_x \sin \theta_z}{\cos \theta_x - \cos \theta_z}\right) \quad (17)$$

A numerical visualization of the image of $\mathbf{R}_{y\bar{x}\bar{z}-y\bar{x}}$ verifies that $\mathbf{R}_{y\bar{x}\bar{z}-y\bar{x}}$ corresponding to θ_y in Eq. (17) are located on the boundary between feasible and infeasible orientations [3].

We plot the time history of θ during the process for solving Eq. (15) in Figs. 5(a) and 5(b) to demonstrate how the Jacobian algorithm fails when the Jacobian becomes singular. Note that θ in

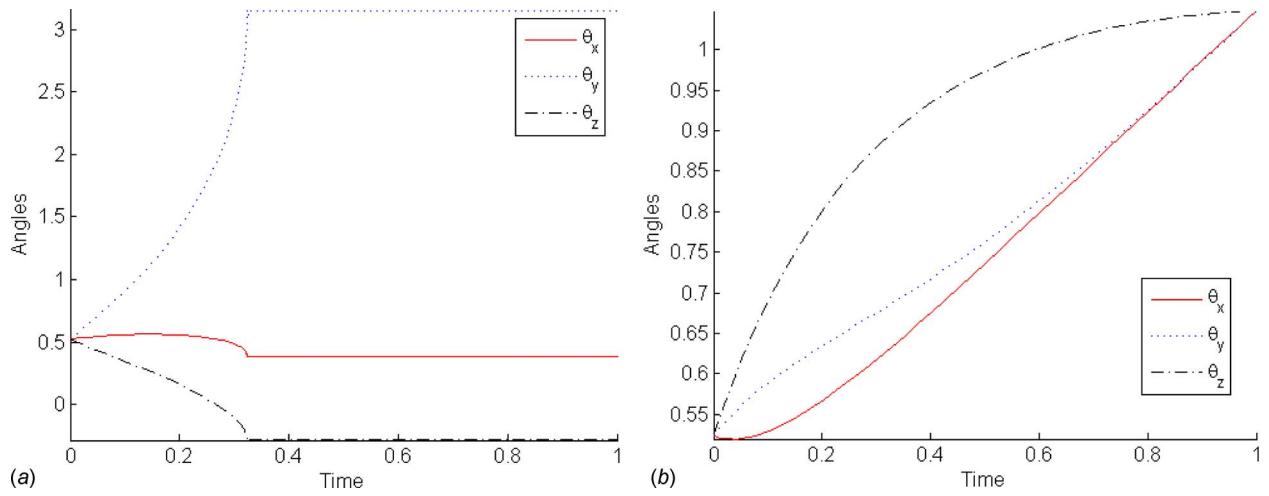


Fig. 5 The angle of rotation $\theta(t) = [\theta_x, \theta_y, \theta_z]$ of (a) R_{xyxzyz} for the desired orientation $R_f = R_{xyxzyz}(\theta_x = -1.9422, \theta_y = 2.1605, \theta_z = -2.0488)$ and (b) R_{yzxzyx} for the desired orientation $R_f = R_{yzxzyx}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$

Figs. 5(a) and 5(b) does not represent how the angle of rotation in the FR rotation varies while the rigid body is being driven toward a desired orientation. After solving Eq. (15), we use the final rotation angle θ_f at $t=1$ to synthesize the desired orientation. Figure 5(a) shows the variation of the rotation angle in the FR rotation R_{xyxzyz} with time during the process for solving Eq. (15). In this example, the Jacobian algorithm is run for the desired orientation $R_f = R_{xyxzyz}(\theta_x = -1.9422, \theta_y = 2.1605, \theta_z = -2.0488)$ from the initial orientation $R_i = R_{xyxzyz}(\theta_x = \pi/6, \theta_y = \pi/6, \theta_z = \pi/6)$. As shown in Fig. 5(a), while the Jacobian algorithm is being run, θ_y approaches $\theta_y = \pi$, where J_{xyxzyz} is singular. Once the Jacobian J_{xyxzyz} becomes singular, the Jacobian program is terminated because J_{xyxzyz} that constructs Eqs. (10) and (15) is not invertible. The Jacobian program shown in Fig. 5(a) was stopped at $t=0.3249$ s. In contrast to R_{xyxzyz} , R_{yzxzyx} covers the entire $SO(3)$ and can attain any possible orientations of a rigid body. Figure 5(b) shows variations in the angle of rotation $\theta = [\theta_x, \theta_y, \theta_z]$ versus time while solving Eq. (15) for a given desired orientation R_f . We solved Eq. (15) from the initial orientation $R_i = R_{yzxzyx}(\theta_x = \pi/6, \theta_y = \pi/6, \theta_z = \pi/6)$ to the final orientation $R_f = R_{yzxzyx}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$. Unlike the example of R_{xyxzyz} shown in Fig. 5(a), the solution to Eq. (15) using R_{yzxzyx} provides the angle of rotation that synthesizes the desired final orientation R_f . This example explains why the FR rotation R_{yzxzyx} is selected for the implementation of the Jacobian-based algorithm.

A number of numerical simulation results verify that the converged angle of rotation is identical with the closed-form solutions to the inverse kinematics problem found by Koh and Ananthasuresh [4], as shown in Fig. 5(b). The closed-form solutions identify the FR sequence that synthesizes the desired orientation in a deterministic way. If the desired orientation $R_f = R_{yzxzyx}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$ is given, the closed-form solutions determine the four-rotation angles in Table 1 that synthesize R_f .

Table 1 Four-rotation angles that synthesize $R_f = R_{yzxzyx}(\theta_x = 60 \text{ deg}, \theta_y = 60 \text{ deg}, \theta_z = 60 \text{ deg})$

Solution No.	θ_x (deg)	θ_y (deg)	θ_z (deg)
1	60	60	60
2	60	-120	120
3	-159.68	-150.7	3.3
4	-159.68	29.29	176.7

In order to examine the relationship between the numerical solutions obtained using the single-Jacobian method and the closed-form solutions to the inverse kinematics problem, we implemented the single-step Jacobian method for 50 random initial orientations and the desired orientation $R_f = R_{yzxzyx}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$. Then, we plotted the trajectories of θ in the space of rotation angles T^3 , as shown in Fig. 6. The four-rotation angles determined by the closed-form solution are represented by squares, and the angle of rotation defining each initial orientation is represented by circles. The straight paths connecting each circle and one of the four squares in Fig. 6 imply that all the numerical solutions of the single-step Jacobian method converge into one of the four FR sequences shown in Table 1. Therefore, this numerical simulation result verifies that the single-step Jacobian method is capable of converging to the analytical solutions regardless of given initial orientations. Both the closed-form solution and the single-step Jacobian method provide a direct way to achieve a desired orientation. However, after finding all the solutions indicated in Table 1, we have to select an FR sequence composed of the smallest rotations among the four solutions for an efficient maneuvering of the rigid body. On the other hand, the single-step Jacobian method determines an FR sequence composed of small rotations just by using an initial orientation determined by small rotation angles. Note, however, that these rota-

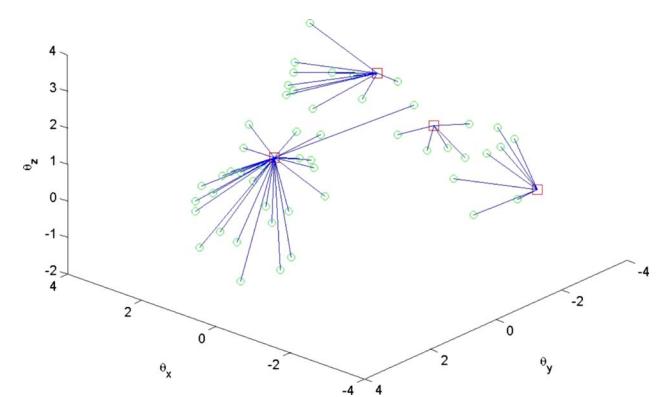


Fig. 6 Trajectories of θ in T^3 converged from 50 random initial orientations. \square represents analytical solutions to the inverse kinematics problem in T^3 . \circ represents the angle of rotation corresponding to the initial orientations provided to the single-step Jacobian algorithm.

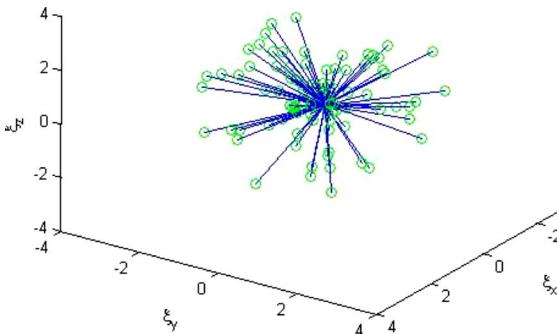


Fig. 7 Trajectories of $\xi = [\log[\mathbf{R}]]^V$ converged into 100 random desired orientations. The random desired orientations are located at the origin, and the initial orientation \mathbf{R}_i is represented by \circ .

tions may still not be small enough for practical implementation, which leads to the motivation for developing the multiple-step Jacobian algorithm that will be discussed in the next section.

In order to examine the convergence property of the single-step Jacobian algorithm, we solve the Jacobian equation in Eq. (15) for 100 random desired orientations and plot the trajectory of $\xi = [\log[\mathbf{R}]]^V \in \mathbb{R}^3$ in log space. The three dimensional space where $\xi \in \mathbb{R}^3$ is plotted is referred to as *log space* due to the fact that $\xi \in \mathbb{R}^3$ is acquired by the logarithm of a rotational displacement. For this numerical simulation, the FR motion $\mathbf{R}_{y_{xz}^{-}y_{-x}}(\theta_x = \pi/6, \theta_y = \pi/6, \theta_z = \pi/6)$ is selected as an initial orientation \mathbf{R}_i and represented by circles in Fig. 7. Although we provide one fixed initial orientation \mathbf{R}_i to Eq. (15), it is distributed at multiple locations in log space because it is translated by the desired orientation so that the converged orientations are located at the origin if the orientation is converged to the desired orientation. Therefore, the identity located at the origin $\xi = \{0, 0, 0\}^T$ represents the random desired orientations. The initial orientation and the converged orientations are connected by a straight line in Fig. 7. Since the single-step Jacobian method achieves the desired orientation in a single FR rotation, there are no intermediate orientations between the initial and converged orientations. The straight lines in Fig. 7 exhibit how the initial orientation \mathbf{R}_i is associated with the converged orientations. The paths converged to the orientations at the origin demonstrate that all the solutions converge to the desired orientations located at the origin and verify the robust convergence property of the single-step Jacobian method.

4.2 Multiple-Step Jacobian Method. In practice, the amount of input rotation provided to a rigid body could be limited depending on the moment of inertia of the system and physical torque-generation-mechanisms mounted on the system. In order to plan practical paths for such systems, we need to plan the motion of the rigid body using only small rotations. We explore a new approach in which a rigid body's orientation is adjusted by the single-step Jacobian method applied multiple times through an incremental optimal path between initial and final orientations. The implementation of the single-step Jacobian method at each step requires initial and desired orientations, which are referred to as local initial and local final orientations, respectively, to distinguish them from global initial and global final orientations that describe the global orientation change of the rigid body along its entire path.

To drive the rigid body through feasible orientations on the optimal path, we divide the entire path into a series of small intervals and maneuver the rigid body such that the desired incremental orientations on the optimal path are achieved at each step. In order to find an FR sequence that reorients the body to \mathbf{R}_f , we solve the Jacobian equation in Eq. (15) for the desired orientation $\mathbf{R}_d = \mathbf{R}_i^T \mathbf{R}_f$. The optimal path in $SO(3)$ between the initial orientation \mathbf{I} and the desired orientation \mathbf{R}_d is constructed by $\exp[\phi \hat{\mathbf{w}}]$

and $\phi \hat{\mathbf{w}} = \log[\mathbf{R}_d]$. The path $\exp[\phi \hat{\mathbf{w}}]$ determined in this way defines the rigid body's optimal orientation change between the initial and desired orientations. In order to discretize the optimal orientation change, which also signifies dividing the optimal path, we divide the optimal path into a series of small intervals using a fixed incremental rotation angle $\Delta\phi$. The incremental desired rotational displacement $\Delta\mathbf{R}_d = \exp[\Delta\phi \hat{\mathbf{w}}]$ is acquired by replacing the rotation angle ϕ in $\phi \hat{\mathbf{w}}$ with the small rotation angle $\Delta\phi$. Then, $\Delta\mathbf{R}_d$ represents a local desired rotational displacement at an interval of the path.

Once $\Delta\tilde{\mathbf{R}}$ that approximates the incremental rotational displacement $\Delta\mathbf{R}_d$ is determined by the single-step Jacobian algorithm using a constant angular velocity $\Delta\boldsymbol{\omega}_{opt} = [\Delta\phi \hat{\mathbf{w}}]^V$, a new desired orientation \mathbf{R}_d^{new} is established for the next step. For a given desired orientation \mathbf{R}_d and approximated rotational displacement $\Delta\tilde{\mathbf{R}}$, the new desired orientation \mathbf{R}_d^{new} required for evaluation of the next desired incremental rotational displacement is $\mathbf{R}_d \Delta \tilde{\mathbf{R}}^T$. Then the incremental rotation for the next step is decided by dividing $\log[\mathbf{R}_d \Delta \tilde{\mathbf{R}}^T]$ in the same manner as for $\Delta\mathbf{R}_d$.

The formulation of the multiple-step Jacobian method is implemented, as shown in the flow chart in Fig. 8. The multiple-step Jacobian algorithm is implemented for a given initial orientation \mathbf{R}_i and final orientation \mathbf{R}_f . In Fig. 8, the Jacobian equation in Eq. (15) is solved for an incremental desired orientation $\Delta\mathbf{R}_d$ until the distance between the desired orientation \mathbf{R}_d and the current orientation \mathbf{R} is less than the tolerance, Tol . The distance between two orientations is measured by the Frobenius matrix norm D defined by

$$D = \|\mathbf{R}_d - \mathbf{R}(t)\| = [\text{trace}\{(\mathbf{R}_d - \mathbf{R}(t))^T (\mathbf{R}_d - \mathbf{R}(t))\}]^{1/2} \quad (18)$$

where $\mathbf{R}(t)$ indicates the rigid body's orientation at time t . If the convergence criterion $\|\mathbf{R}_d - \mathbf{R}\| < Tol$ is not satisfied, the Jacobian equation is solved for a new desired orientation \mathbf{R}_d^{new} to drive the rigid body further toward the given desired orientation \mathbf{R}_d .

The convergence property of the multiple-step Jacobian method is explored through numerical simulations that are carried out for a number of randomly selected desired orientations. In Fig. 9, the trajectory of $\xi = [\log[\mathbf{R}]]^V \in \mathbb{R}^3$ converged to 100 random orientations are plotted in log space. In this example, $\mathbf{R}_i = \mathbf{R}_{y_{xz}^{-}y_{-x}}(\theta_x = \pi/6, \theta_y = \pi/6, \theta_z = \pi/6)$ is chosen as an initial orientation and is translated such that a desired orientation is located at the origin of the log space.

Therefore, the origin in Fig. 9 represents the global desired orientation \mathbf{R}_f in all numerical simulations. Although we use only one fixed global initial orientation, it is distributed in many different locations represented by circles, as it is translated so as to be located at the origin if the orientation converges to the desired orientation. As the multiple-step Jacobian method reaches the desired orientation through incremental desired orientations located on the optimal path, the path connecting the initial orientation and the converged orientations in Fig. 9 exhibits a real trajectory of the rigid body in log space. Figure 9 shows that all the numerical solutions of the multiple-step Jacobian method converged to the 100 random global desired orientations. This numerical simulation result demonstrates the robust convergence ability of the multiple-step Jacobian algorithm.

The features of the multiple-step and the single-step Jacobian methods are clearly captured when their distance errors are compared. Figures 10(a) and 10(b) show the distance error of the single-step and the multiple-step Jacobian algorithms in time, respectively, when both are implemented for the desired orientation $\mathbf{R}_{y_{xz}^{-}y_{-x}}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$. The monotonically decreasing tendency of the distance error of the multiple-step Jacobian method, shown in Fig. 10(b), demonstrates that the orientation traverses the optimal path. In contrast, the distance error of the single-step Jacobian method shown in Fig. 10(a) shows that

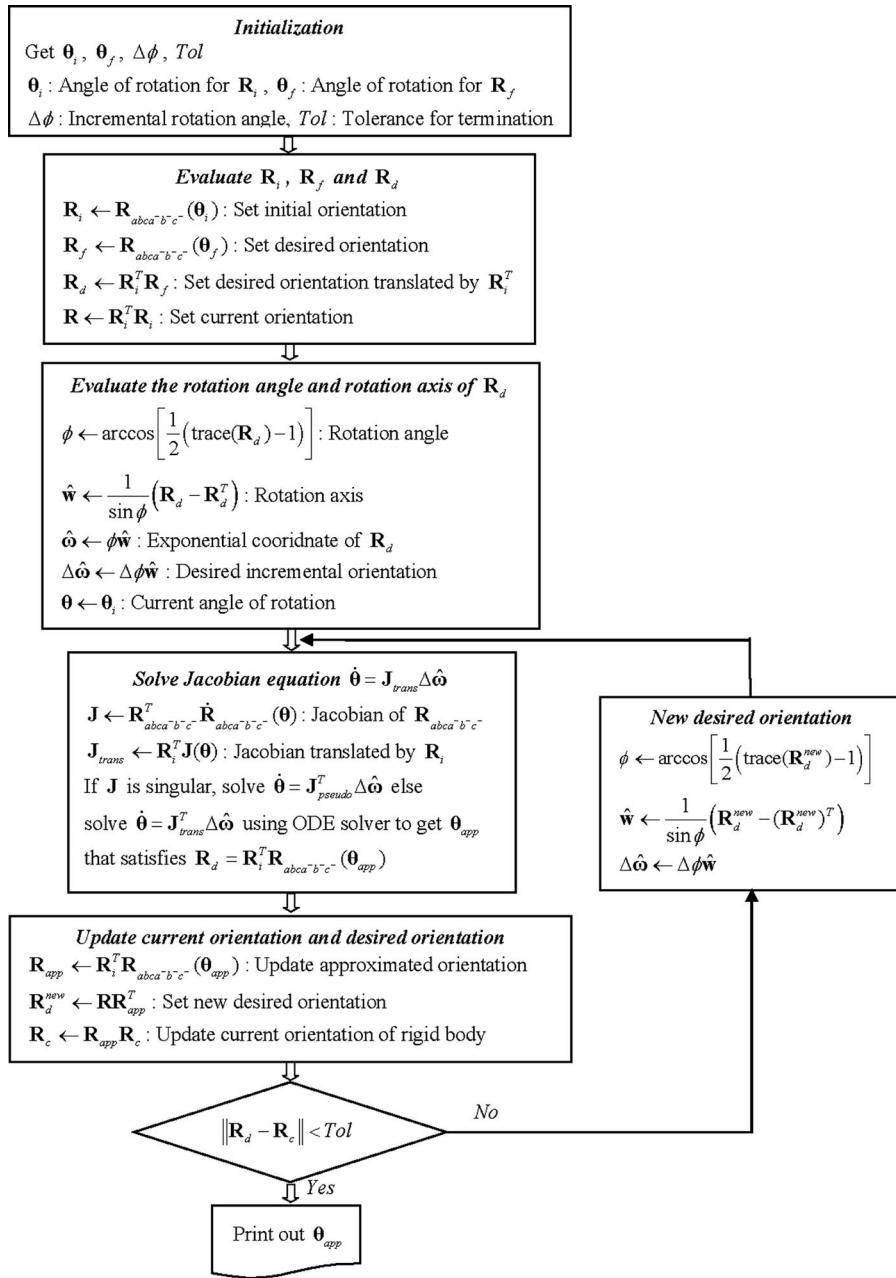


Fig. 8 Flow chart of the multiple-step Jacobian-based algorithm

the path constructed by the solution of the single-step Jacobian method is not optimal as it does not show a monotonically decreasing behavior.

Although the single-step Jacobian method does not adjust orientation along the optimal path and may require large rotations to achieve the desired orientation, it is more efficient than the multiple-step Jacobian method in the sense that the single-step Jacobian method reaches the desired orientation through a shorter rotational path. The rotational path length L used in this paper is defined as

$$L = \int_0^{t_f} \sum_{i=1}^6 |\theta_i| dt \approx \sum_{k=1}^{N_t} \sum_{i=1}^6 |\theta_i^k| \Delta t_k \quad (19)$$

where $|\theta_i|$ is the angle of the i th rotation in the FR sequence of rotations and $|\theta_i^k|$ represents the angle of the i th rotation at the k th time interval. The integration in Eq. (19) is approximated as a

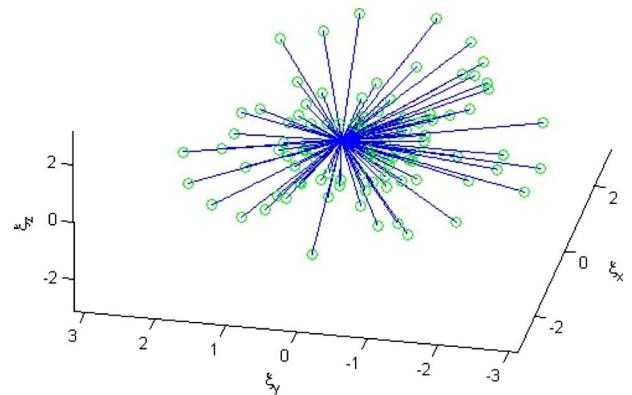


Fig. 9 Trajectories of ξ converging into 100 random desired orientations. \circ represents an initial orientation \mathbf{R}_i , and random desired orientations are located at the origin.

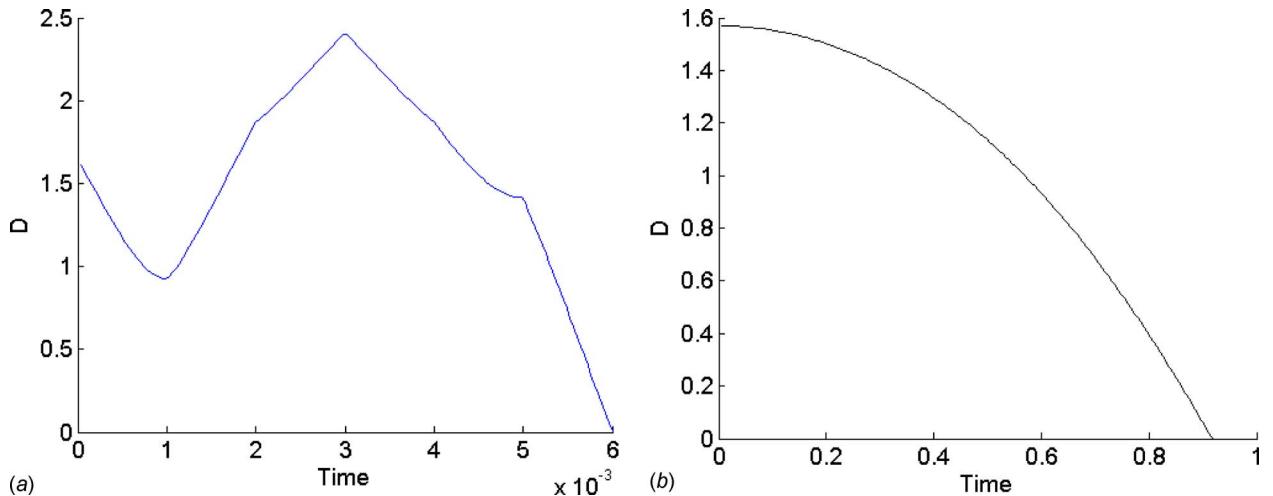


Fig. 10 Distance error D of (a) the single-step and (b) the multiple-step Jacobian methods that are run for the desired orientation $\mathbf{R}_{y_{zxz}-y-x}(\theta_x=\pi/3, \theta_y=\pi/3, \theta_z=\pi/3)$

Riemann sum of the angle of rotation over a discrete time domain [14]. As mentioned in Sec. 1, we assume that each rotational operation in FR rotations is completed over the time period $\Delta t_k = 1 \times 10^{-3}$ s.

In order to investigate the efficiency of the single- and the multiple-step Jacobian algorithms, we compare the rotational path length of the single-step Jacobian algorithm with that of the multiple-step Jacobian algorithm that accomplishes the same task over two steps. This numerical experimentation will demonstrate which algorithm takes a longer rotational path to complete identical tasks. It has already been demonstrated that the single-step Jacobian algorithm identifies exact solutions to the inverse kinematics problem discussed in Sec. 4.1. If $\mathbf{R}_{y_{zxz}-y-x}(\theta_x=\pi/3, \theta_y=\pi/3, \theta_z=\pi/3)$ is selected as a desired orientation \mathbf{R}_f , it is apparent that the angle of rotation that synthesizes the desired orientation \mathbf{R}_f over a single step is $\{\theta_x=\pi/3, \theta_y=\pi/3, \theta_z=\pi/3\}^T$. For a fair comparison of the two control algorithms, the multiple-step Jacobian algorithm is run over two steps for the identical desired orientation \mathbf{R}_f . The desired incremental rotation for the first step is decided such that the incremental rotation angle is half of the total rotation angle of the desired orientation change \mathbf{R}_f so that the desired orientation can be achieved over two steps. As the total rotation angle ϕ of the desired orientation $\mathbf{R}_{y_{zxz}-y-x}(\theta_x=\pi/3, \theta_y=\pi/3, \theta_z=\pi/3)$

$=\pi/3, \theta_z=\pi/3$ is $\phi_f=1.2447$ rad, the intermediate desired orientation $\mathbf{R}_{f/2}$ are determined by $\exp[\phi_{f/2}\mathbf{w}]$, where $\phi_{f/2}=\phi_f/2=0.6223$ rad. According to the closed-form solutions [4], the angles of rotation that achieve the intermediate orientation $\mathbf{R}_{f/2}$ is $\{\theta_x=0.5487, \theta_y=0.5234, \theta_z=1.1598\}^T$ and the same angles of rotation are required to cover the remaining path to the desired destination \mathbf{R}_f .

The relative path length of the Jacobian-based algorithms is evaluated using the rotational path length defined in Eq. (19). As shown in Fig. 11(a), it turns out that the multiple-step Jacobian algorithm run over two steps takes a longer rotational path than the single-step Jacobian algorithm. While the rotational path length of the single-step algorithm is 0.0377 rad s, the multiple-step Jacobian algorithm takes a longer path 0.0536 rad s to converge to the same destination over two steps. This result implies that the Jacobian-based algorithm tends to take a longer rotational path when it is run over multiple steps along an incremental path to the same desired orientation. This feature becomes more apparent when the single-step Jacobian method is compared with the multiple-step Jacobian method run for an incremental rotation angle. In Fig. 11(b), the rotational path length of the multiple-step algorithm, which is run for the incremental rotation angle $\Delta\phi$

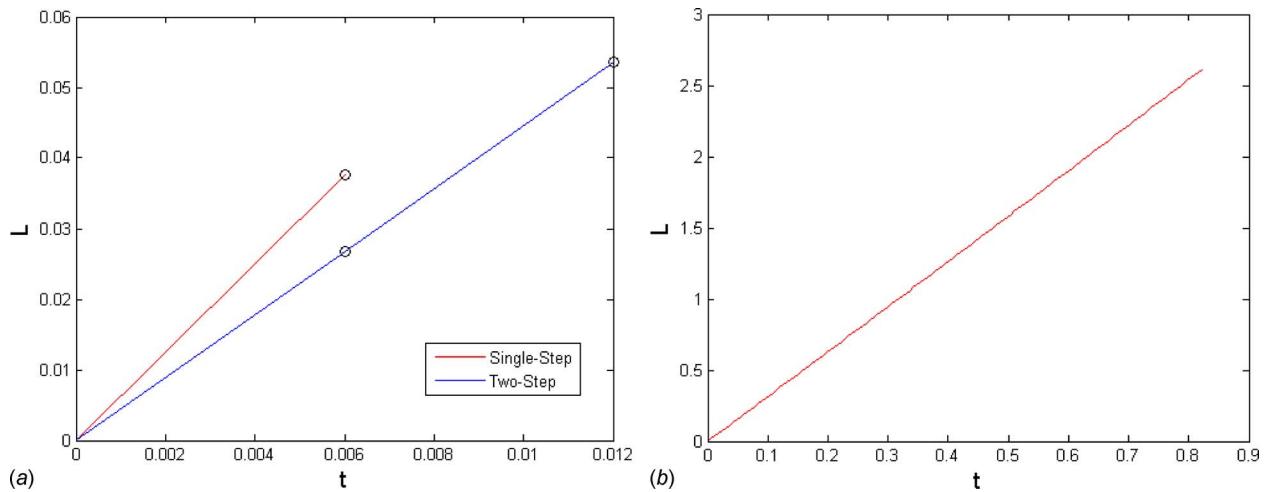


Fig. 11 (a) Rotational path length L of the single-step Jacobian, two-step Jacobian; (b) multiple-step Jacobian methods ($\Delta\phi=0.0001$ rad) that are run for the desired orientation $\mathbf{R}_{y_{zxz}-y-x}(\theta_x=\pi/3, \theta_y=\pi/3, \theta_z=\pi/3)$

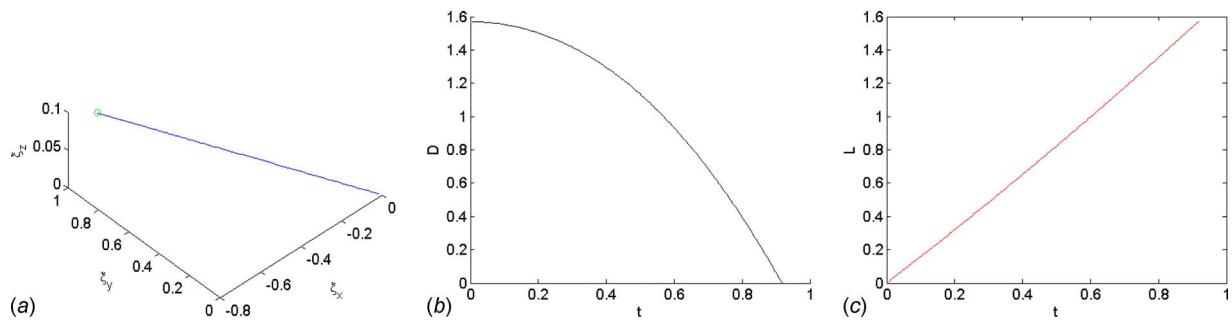


Fig. 12 Multiple-step Jacobian method: (a) trajectory of ξ , (b) distance error D , and (c) rotational path length L

$=0.0001$ rad and $\mathbf{R}_f = \mathbf{R}_{y_{xz}^{-}y^{-}x^{-}}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$, is 2.68 rad s. This consequence clearly shows that the multiple-step Jacobian algorithm takes a longer rotational path than the single-step algorithm that travels 0.0377 rad s to reach the same desired orientation, as shown in Fig. 11(a). This numerical analysis indirectly signifies that the single-step Jacobian algorithm could be more efficient than the multiple-step Jacobian algorithm depending on the mechanisms of the system because the single-step Jacobian algorithm takes a shorter rotational path to a desired orientation. However, the single-step Jacobian method may require large rotations to reach a desired destination while the multiple-step Jacobian method is capable of maneuvering a rigid body using feasible small rotations. Therefore, the multiple-step Jacobian algorithm suggests an alternative and practical way of maneuvering a rigid body through fully-reversed motions when the amount of input rotation is limited and hence the rigid body needs to be manipulated via only small rotations.

5 Comparison With the Pair-Wise and Single-Sequence Algorithms

In this section, we investigate the performance of the multiple-step Jacobian algorithm by comparing it with existing motion planning algorithms developed for fully-reversed rotations. Koh et al. [2] developed the pair-wise and single-sequence algorithms based on the leading order approximation of fully-reversed rotations in Lie algebra $so(3)$. In order to explore the performance of these algorithms, their convergence rate and the rotational path length are compared. Comparisons of convergence rates provide an estimation of how rapidly a numerical algorithm converges to desired orientations, and the rotational path length indirectly reveals which motion planning algorithm is more efficient. We carried out a number of numerical simulations under identical conditions to compare the performance of these algorithms.

FR sequences of rotations can be expanded in a series form on $so(3)$. Then, the net orientation change of a rigid body by fully-reversed infinitesimal rotations can be approximated using the leading order terms in the series [2]. The first order effect of FR infinitesimal rotations is nullified as infinitesimal rotations are

commutative. Therefore, the orientation changes by fully-reversed infinitesimal rotations are governed by the leading second order terms in its series expanded by the Campbell–Baker–Hausdorff formula [2,15]. The orientation of the rigid body is adjusted by either pair-wise rotations or single-sequence rotations. The pair-wise rotation refers to four-rotation FR rotations, and the single-sequence rotation refers to six-rotation FR rotations. A four-rotation FR sequence of rotations is composed of initial two rotations about the axes of the body-fixed coordinate frame and subsequent two rotations that undo the proceeding two rotations. According to its leading order effect, FR rotations about two body-fixed axes, $\mathbf{R}_a(\theta_a)\mathbf{R}_b(\theta_b)\mathbf{R}_a(-\theta_a)\mathbf{R}_b(-\theta_b)$, are equivalent to a rotation about the remaining third axis, $\mathbf{R}_c(\theta_a\theta_b)$. The pair-wise algorithm makes use of the fact that Euler rotations such as $\mathbf{R}_a\mathbf{R}_b\mathbf{R}_c$ can achieve any orientations in $SO(3)$. Each rotation constituting the Euler rotation is approximated by the corresponding pair-wise rotation. Therefore, the pair-wise algorithm performs 12 successive rotations to reach an incremental desired orientation. On the other hand, the single-sequence algorithm approximates the desired orientation using six-rotation FR sequences of rotations. While the pair-wise algorithm is featured by its robust convergence capability, the single-sequence algorithm accomplishes the same task in an efficient manner as it performs only six rotations to reach the same incremental desired orientation [2].

For a fair comparison of the these motion planning algorithms, the numerical simulations of the multiple-step Jacobian, pair-wise, and single-sequence algorithms are undertaken for the same global desired orientations using the same incremental rotation angles on a single-processor desktop computer. For a global desired orientation \mathbf{R}_f , a local incremental desired orientation $\Delta\mathbf{R}_f$ is established using a fixed incremental rotation angle $\Delta\phi$ at each step. For all examples presented in this section, the orientation $\mathbf{R}_{y_{xz}^{-}y^{-}x^{-}}(\theta_x = \pi/3, \theta_y = \pi/3, \theta_z = \pi/3)$ is provided as a desired orientation. Although we have performed a number of numerical simulations for various incremental rotation angles to explore the performance of these motion planning algorithms, here we present an example in which the incremental rotation angle $\Delta\phi$

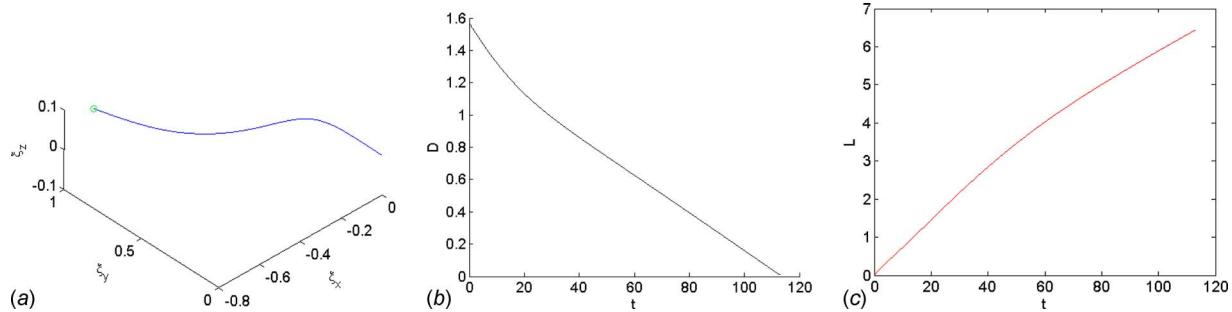


Fig. 13 Pair-wise method: (a) trajectory of ξ , (b) distance error D , and (c) rotational path length L

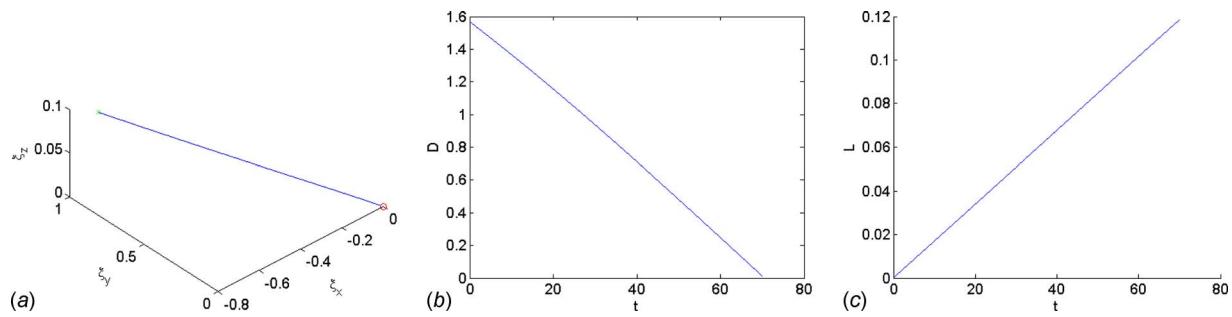


Fig. 14 Single-sequence method: (a) trajectory of ξ , (b) distance error D , and (c) rotational path length L

$=0.0001$ rad is used. Other simulations carried out for various other incremental rotation angles also show the features highlighted in this example.

The characteristics and strengths of the multiple-step Jacobian, pair-wise, and single-sequence methods are investigated by comparing the trajectory of ξ , distance error, and rotational path length while the rigid body is being driven toward the desired destination. In Figs. 12(a), 13(a), and 14(a), the trajectory from the initial to the final orientations is drawn in log space to explore how the orientation travels in log space. While the trajectories by the multiple-step method and the single-sequence method appeared as a straight line (Figs. 12(a) and 14(a)), the trajectory determined by the pair-wise method takes a curved path, as shown in Fig. 13(a). This implies that the pair-wise algorithm takes a longer path than the multiple-step and single-sequence algorithms.

Comparing the time cost for convergence in Table 2, the pair-wise method takes 112.8840 s while the multiple-step method spends 0.9108 s and the single-sequence method costs 70.02 s to converge to the desired orientation. As mentioned in the Introduction, it is assumed that it takes 6 ms for completing a six-rotation FR sequence as one rotation in FR motion is accomplished in 1 ms. This result demonstrates that the pair-wise method takes more time than the other two algorithms for convergence. The pair-wise algorithm performs 12 successive rotations at each step to achieve incremental desired local orientations while the multiple-step and single-sequence methods require six rotations to accomplish the same task. As the pair-wise algorithm performs more rotations at each step than the other two algorithms, it usually takes a greater time to reach the same desired orientation compared to the multiple-step and single-sequence algorithms. The rotational path lengths of each numerical simulation are compared in Table 2 and Figs. 12(c), 13(c), and 14(c) suggest that the pair-wise method takes the longest rotational path among all the three algorithms.

According to the plots and data in Figs. 12(b), 13(b), and 14(b), and in Table 2, the multiple-step Jacobian method shows a faster convergence rate than the other two control algorithms. This fast convergence rate of the multiple-step method is attributed to the feature of the single-step Jacobian method that identifies exact desired orientations on the path. The pair-wise and single-sequence methods approximate desired orientations based on the second order effect of FR sequences of rotations. In contrast, a rigid body maneuvered by the multiple-step Jacobian method moves through exact optimal intermediate orientations at each

step, which makes the Jacobian-based algorithm take fewer steps than the pair-wise and single-sequence algorithms to reach the desired orientation. The data shown in Table 2 confirms that (i) the multiple-step algorithm possesses the fastest convergence property among the three algorithms, and that (ii) the pair-wise method requires the most number of steps for convergence and costs the most amount of time.

6 Conclusions

We propose two motion planning algorithms that are based on the Jacobian of the forward kinematics mapping of fully-reversed rotations. The single-step Jacobian algorithm determines a specific FR sequence of rotation that synthesizes a desired orientation and is also one of the closed-form solutions to the inverse kinematics problem. While the closed-form solutions [4] require not-necessarily-small rotations to realize a given desired orientation, the single-step Jacobian algorithm identifies an FR sequence that is composed of the smallest rotations among the four analytical solutions without intensive computation. The multiple-step Jacobian method enables maneuvering a rigid body along an optimal path with a series of feasible rotations determined by the single-step Jacobian method. The robust convergence ability of the single- and multiple-step Jacobian methods was demonstrated through a number of numerical simulations run for random desired orientations. The fast convergence property of the multiple-step Jacobian algorithms was verified in comparison with the pair-wise and single-sequence algorithms on a single-processor desktop computer. Therefore, it is preferable to use the Jacobian algorithm if a rapid manipulation of a rigid body is required.

In summary, the Jacobian algorithm provides a fast and convenient means of planning maneuvers for a rigid body undergoing FR rotations. We would like to stress that these features of the Jacobian algorithm allow it to be more useful than the pair-wise and single-sequence algorithms depending on given tasks. The Jacobian algorithm performs better than the existing algorithms when a fast manipulation of a vehicle is required. In the future, we will explore the possibility of using the Jacobian-based algorithm for motion planning for a rigid body undergoing four-rotation FR sequence of rotations. The algorithm presented here may also be useful to other problems where the Jacobian is a singular or null matrix.

Acknowledgment

The financial support of Professor Wankyun Chung at Pohang University of Science & Technology is gratefully acknowledged.

References

- [1] Li, J., Koh, S. K., Ananthasuresh, G. K., and Ananthakrishnan, S., 2001, "A Novel Attitude Control Technique for Miniature Spacecraft," *MEMS Symposium, Vol. 1 CD-ROM Proceedings of the MEMS Symposium at the 2001 ASME International Mechanical Engineering Conference and Exposition*, New York, Nov. 11–16.
- [2] Koh, S. K., Ostrowski, J. P., and Ananthasuresh, G. K., 2002, "Control of Micro-Satellite Orientation Using Bounded-Input, Fully-Reversed MEMS Ac-

Table 2 Numerical simulation results obtained for $\Delta\phi = 0.0001$ rad

Criterion	Multiple-step	Pair-wise	Single-sequence
Time cost (s)	0.9108	112.8840	70.02
Rotational path length (rad s)	1.5709	6.4312	0.1184

- tuators," *Int. J. Robot. Res.*, **21**(5–6), pp. 591–605.
- [3] Koh, S. K., Ananthasuresh, G. K., and Croke, C., 2004, "Analysis of Fully-Reversed Sequences of Non-Commutative Free-Body Rotations," *ASME J. Mech. Des.*, **126**(4), pp. 609–616.
- [4] Koh, S. K., and Ananthasuresh, G. K., 2004, "Inverse Kinematics of an Un-tethered Rigid Body Undergoing a Sequence of Forward and Reverse Rotations," *ASME J. Mech. Des.*, **126**(5), pp. 813–821.
- [5] Bharadwaj, S., Osipchuk, M., Mease, K. D., and Park, F. C., 1998, "Geometry and Inverse Optimality of Global Attitude Stabilization," *J. Guid. Control Dyn.*, **21**(6), pp. 930–939.
- [6] Bloch, A. M., Krishnaprasad, P. S., Marsden, J. E., and Sanchez de Alvarez, G., 1992, "Stabilization of Rigid Body Dynamics by Internal and External Torques," *Automatica*, **28**, pp. 745–756.
- [7] Bullo, F., Murray, R. M., and Sarti, A., 1995, "Control on the Sphere and Reduced Attitude Stabilization," *Nonlinear Control Systems Design Symposium*, Also Technical Report CIT/CDS 95-005, available electronically via <http://avalon.caltech.edu/cds>
- [8] Koditschek, D. E., 1989, "The Application of Total Energy as a Lyapunov Function for Mechanical Control Systems," *Dynamics and Control of Multi-body Systems*, P. S. Krishnaprasad, J. E. Marsden, and J. C. Simo, eds., AMS, Providence, RI, Vol. 97, pp. 131–157.
- [9] Wen, J. T.-Y., and Kreutz-Delgado, K., 1991, "The Attitude Control Problem," *IEEE Trans. Autom. Control*, **36**(10), pp. 1148–1162.
- [10] Murray, R. M., Li, Z., and Sastry, S. S., 1993, *A Mathematical Introduction to Robotic Manipulation*, CRC, Boca Raton, FL.
- [11] Stein, D., Scheinerman, E. R., and Chirikjian, G. S., 2003, "Mathematical Models of Binary Spherical-Motion Encoders," *IEEE/ASME Trans. Mechantron.*, **8**(2), pp. 234–244.
- [12] Chirikjian, G. S., and Kyatkin, A. B., 2000, *Engineering Application of Non-commutative Harmonic Analysis*, CRC, Boca Raton, FL.
- [13] 2004, MATLAB, Numerical Analysis Software from Mathworks, Inc. Woburn, MA, www.mathworks.com
- [14] Marsden, J. E., and Hoffman, M. J., 1993, *Elementary Classical Analysis*, W. H. Freeman and Company, New York.
- [15] Sastry, S., and Marsden, J. E., 2004, *Nonlinear Systems: Analysis, Stability and Control*, Springer-Verlag, New York.