

Research Article

Geometric Pseudospectral Method on SE(3) for Rigid-Body Dynamics with Application to Aircraft

Jie Li, Honglei An, Huayong Zhu, Lincheng Shen, and Bin Fang

College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Huayong Zhu; o_fantasy_o@163.com

Received 29 December 2012; Accepted 26 March 2013

Academic Editor: Anders Eriksson

Copyright © 2013 Jie Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

General pseudospectral method is extended to the special Euclidean group SE(3) by virtue of equivariant map for rigid-body dynamics of the aircraft. On SE(3), a complete left invariant rigid-body dynamics model of the aircraft in body-fixed frame is established, including configuration model and velocity model. For the left invariance of the configuration model, equivalent Lie algebra equation corresponding to the configuration equation is derived based on the left-trivialized tangent of local coordinate map, and the top eight orders truncated Magnus series expansion with its coefficients of the solution of the equivalent Lie algebra equation are given. A numerical method called geometric pseudospectral method is developed, which, respectively, computes configurations and velocities at the collocation points and the endpoint based on two different collocation strategies. Through numerical tests on a free-floating rigid-body dynamics compared with several same order classical methods in Euclidean space and Lie group, it is found that the proposed method has higher accuracy, satisfying computational efficiency, stable Lie group structural conservativeness. Finally, how to apply the previous discretization scheme to rigid-body dynamics simulation and control of the aircraft is illustrated.

1. Introduction

The aircraft is usually regarded as a single rigid body in three-dimensional space. The dynamics of a rigid body is an important problem modeling the time evolution of aircraft and other vehicles [1]. In particular, in aircraft simulation and control, its authenticity affects the fidelity of the simulation and also determines whether it can truly reflect the performance of the designed controller to a certain extent. The rigid-body dynamics has fundamental motion invariants; for example, the flow of a Hamiltonian system is symplectic [2, 3], the total energy of a rigid body is conserved in the absence of nonconservative forces [4], and the momentum map associated with a symmetry of a rigid body is preserved [5]. Furthermore, the exact flow of a rigid body always stays on a configuration manifold, since the combined configurations of the translation and rotation of the rigid body construct a special Lie group called the special Euclidean group SE(3) [6]. The invariants often manifest through geometric characteristics of exact flow of a rigid

body, such as symplecticity, first integrals, symmetry, and Lie group structure. However, for most rigid bodies, obtaining the analytical solution is quite difficult. Thus, under the condition of non analytical solution, it is desired to develop a numerical method whose iterates preserve the previous fundamental invariants [7]. For this purpose, the research area of development of numerical methods for rigid-body differential equations that preserve geometric properties of the numerical solution has been of great interest in recent years [8]. Simos [8, 9], Monovasilis et al. [3, 10], and Kalogiratos [11] construct several different multistep symplectic integrators based on symplectic geometry in order to preserve the Hamiltonian energy of rigid bodies as the harmonic oscillator, the pendulum, two-body problem, and orbital problem. Lee et al. [5] and Hussein et al. [12] adopt variational integrators to evolve the rigid-body dynamics, which ensure that while the total energy of the rigid body is conserved under conservative forces, the momentum associated with a symmetry of the system is also conserved. Iserles et al. [13] and Lee et al. [5] extend numerical methods in Euclidean

space into Lie group, so that numerical flow of the rigid-body dynamics has Lie group structural conservativeness, which is also the main focus of this paper. Retaining motion invariants under discretization has been proven not only a nice mathematical property, but also the key to improved numerics, as they capture the right dynamics (even in long-time integration) and exhibit increased accuracy [2, 7, 13, 14]. Therefore, for driving the evolution of rigid-body dynamics, it is very important to develop a numerical method with preservation of the previous geometric characteristics.

The difficulty of developing a numerical method on Lie group arises as the group is not the well-known Euclidean space \mathbb{R}^n ; consequently, the rigid-body dynamics evolving on $SE(3)$ cannot be properly integrated by conventional numerical method, including the popular Runge-Kutta schemes, developed for \mathbb{R}^n . For this reason, one resorts to other means to drive the rigid-body dynamics forward while preserving the Lie group structure of the configuration. It can be roughly divided into two categories: classical Lie group methods which discretize the continuous equations of motion and Lie group variational integrators which discretize the variational principles of mechanics. Classical Lie group methods include the Munthe-Kaas (MK) method [15], the Crouch-Grossmann (CG) method [16], the Newmark-type method [17], and the commutator-free (CF) method [18], and so forth. The MK method is based on a differential equation on the Lie algebra and uses a single evaluation of the exponential map. The CG method updates the group elements by multiple evaluations using the exponential map. For detailed information, one can consult to [13]. Lie group variational integrators are proposed in recent years whose idea is to discretize the variational principles of mechanics: Hamilton's principle or Lagrange-d'Alembert principle [1, 5, 14]. Both methods have their own advantage and disadvantage. The former has a better time performance, but its accuracy is not an advantage under the same order condition. The latter preserves exactly energy, momentum, symplectic structure, and group structure [12, 19] and offers a particularly robust and efficient framework for simulation [14], attitude control [4], motion control [1], and trajectory generating [14] of the aircraft and other vehicles, however; it has no advantage of accuracy, and its performance is subjected to the Newton-type solver for solving the equivalent vector equation [5, 14, 20]. Regardless of the classical Lie group methods or Lie group variational integrators, they are all based on Kenth Engø's idea about equivariant map which transforms the differential equations evolving on a Lie group into an equivalent differential equation evolving on a Lie algebra corresponding to the Lie group [21].

Since accuracy, time performance, conservativeness, and numerical stability of the aircraft rigid-body dynamics need to be considered comprehensively in aircraft simulation and control, we do not intend to adopt any of the previous methods but use the same idea of equivariant map to develop a new method on $SE(3)$ for driving the evolution of aircraft dynamics. We resort to pseudospectral method, which is widely used in fluid mechanics, quantum mechanics, linear and nonlinear waves, aerospace, and other fields by virtue of its high accuracy, spectral (or exponential) convergence rates, requirement for less computer memory under the same

precision condition, and so forth [22]. Furthermore, it is commonly used in optimal control of the aircraft and other vehicles [23] and thereby lay the foundation for our simulation and control application. Unlike variational integrator must analytically derive the first-order discrete necessary optimality, which, more often than not, is nontrivial for most optimal control problems [23], typical pseudospectral method can be used to transcribe a continuous optimal control problem into a discrete nonlinear programming problem (NLP), and it has been shown that solving the NLP derived from the pseudospectral transcription of the Gauss form is exactly equivalent to solve the discrete first-order necessary conditions [23, 24]. However, when applied to rigid body dynamics directly, general pseudospectral method cannot preserve Lie group structure. For this purpose, a numerical method on $SE(3)$ called geometric pseudospectral method is proposed in this work, which provides satisfactory accuracy, computational efficiency and preserves the essential Lie group structure. To our knowledge, Moulla et al. [25] was the first to pose the concept of "geometric pseudospectral method" not long ago. They suggested a polynomial pseudospectral method that preserves the geometric structure of port-Hamiltonian systems, the phenomenological laws, and the conservation laws without introducing any uncontrolled numerical dissipation. However, their method was designed only for port-Hamiltonian systems having a special structure, that is, the Dirac structure, thus it could not be directly extended to the systems on $SE(3)$. Thereby, how to apply the pseudospectral method to a dynamics system on $SE(3)$ and preserve essential Lie group structure of the system is still an open question, which is the main topic in this paper.

In this work, we establish a completely rigid-body dynamics model of aircraft in body-fixed frame on $SE(3)$. With respect to kinematics, due to the fact that applying the general pseudospectral method directly to the configuration equation of the rigid-body dynamics could not preserve the Lie group structure of the solution of the equation, drawing on the equivariant map, we transform the configuration equation on $SE(3)$ into an equivalent equations in a Lie algebra space and accordingly give the top eight orders reduced truncated Magnus series expansion with its coefficients $\alpha(\kappa)$ (refer to the appendix) of the solution of the equivalent equation under left trivialization. For the condition of each term in $u^{[2q]}(t)$ being multivariate quadrature, we apply the general pseudospectral method to compute quadrature weights, so that obtaining the values of $u^{[2q]}(t)$ at the collocation points and the endpoint (collectively referred to discrete points). Finally, we compute configurations at the discrete points via coordinate map. It is worth mentioning that the theories with respect to Lie group methods in the extensive literatures [13–15] are almost derived under right trivialization and thus are not suitable for our method, since our modeled aircraft configuration model is left invariant. To this end, we provide a relatively complete set of left trivialization framework for developing geometric pseudospectral method on $SE(3)$. With respect to dynamics, considering the Lie algebra space is isomorphism to \mathbb{R}^n , we use the general pseudospectral method directly to compute the velocities at the same discrete points as that of configuration.

Under the same 4th-order and the same large step size, the proposed method is compared with implicit RK, explicit RKMK, and Gauss pseudospectral method. Through a comprehensive comparison of accuracy, computational efficiency, and Lie group structural conservativeness, the proposed method has higher accuracy, satisfying computational efficiency, stable Lie group structural conservativeness. The proposed method is coordinate-free, no need to switch chart and special handling of singularities. Finally, we give a way of applying the proposed numerical method to rigid-body dynamics simulation and control of the aircraft.

The rest of the paper is organized as follows. In Section 2, a completely left invariant rigid-body dynamics model of the aircraft on SE(3) is established. Geometric pseudospectral method on SE(3) is developed in Section 3. In Section 3.1, for completeness, the basic idea of the general pseudospectral method in Euclidean space is roughly introduced. In Section 3.2, geometric pseudospectral method on SE(3) is developed under left trivialization, and a 4th-order geometric pseudospectral algorithm is given. In Section 3.3, numerical tests are carried out on a free-floating rigid-body model in comparison with four other numerical methods to validate numerical accuracy, computational efficiency, and structural conservativeness of the proposed method. Then, how to apply the proposed method to rigid-body dynamics simulation and control of the aircraft is illustrated in Section 4. Finally, the conclusions and future work are outlined in Section 5.

2. Aircraft Rigid-Body Dynamics Model on SE(3)

The special Euclidean group SE(3) is the semidirect product of SO(3) and \mathbb{R}^3 , whose group element $g_{ab} = (R_{ab}, p_{ab})$ consists of rotational component $R_{ab} \in \text{SO}(3)$ and translational component $p_{ab} \in \mathbb{R}^3$ of the body-fixed frame $\{b\}$, relative to the space frame $\{a\}$, where $\text{SO}(3) = \{R_{ab} \in \mathbb{R}^{3 \times 3} \mid R_{ab} R_{ab}^T = I, \det R_{ab} = +1\}$ is called the special orthogonal group [6]. In this section, we will regard the aircraft as a single rigid body in three-dimensional space and establish a rigid-body dynamics model of the aircraft on SE(3).

2.1. Kinematics Model. The navigation equations of the aircraft are given by [26]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{ab}(t) \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (1)$$

where, \dot{x} , \dot{y} , and \dot{z} are the north, east, and vertical components of the aircraft velocity, respectively, in the locally level geographic frame on the surface of the Earth, u , v , and w denote X -axis component, Y -axis component, and Z -axis component of the aircraft airspeed, respectively, in the body-fixed frame, and $R_{ab}(t)$ is a sequence of ZYX Euler rotations from the body-fixed frame to the locally level geographic frame, given by

$$R_{ab}(t) = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \cos \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (2)$$

Denote $[x, y, z]^T$ and $[u, v, w]^T$ by $p_{ab}^a(t)$ and $v_{ab}^b(t)$, respectively, then we have

$$\dot{p}_{ab}^a(t) = R_{ab}(t) \cdot v_{ab}^b(t). \quad (3)$$

Also, the kinematic equations are given by [26]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + \tan \theta (q \sin \phi + r \cos \phi) \\ q \cos \phi - r \sin \phi \\ \frac{q \sin \phi + r \cos \phi}{\cos \theta} \end{bmatrix}, \quad (4)$$

where, ϕ , θ , and ψ are roll angle, pitch angle, and yaw angle, respectively, which are commonly referred to as Euler angles; p , q , and r denote the roll rate, pitch rate, and yaw rate, respectively, about X -body axis, Y -body axis, and Z -body axis. Then, we have

$$\begin{aligned} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin \theta \\ \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta \\ -\dot{\theta} \sin \phi + \dot{\psi} \cos \phi \cos \theta \end{bmatrix}. \end{aligned} \quad (5)$$

Denote $[p, q, r]^T$ by $\omega_{ab}^b(t)$, and bring in an operator $\hat{\cdot} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$, for all $y \in \mathbb{R}^3$, satisfying $\hat{\omega}_{ab}^b(t) \cdot y = \omega_{ab}^b(t) \times y$, where \times is vector cross-product; we have $\hat{\omega}_{ab}^b(t) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}$ and obtain

$$\dot{R}_{ab}(t) = R_{ab}(t) \cdot \hat{\omega}_{ab}^b(t). \quad (6)$$

Therefore, from the preceding equation and (3), we can construct a kinematics model (configuration) $\dot{\bar{g}}_{ab}(t) = \bar{g}_{ab}(t) \cdot \bar{V}_{ab}^b(t)$, namely,

$$\begin{bmatrix} \dot{R}_{ab}(t) & \dot{p}_{ab}^a(t) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{ab}(t) & p_{ab}^a(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega}_{ab}^b(t) & v_{ab}^b(t) \\ 0 & 0 \end{bmatrix}, \quad (7)$$

where $\bar{g}_{ab}(t)$ and $\bar{V}_{ab}^b(t)$ are, respectively, called the homogeneous representation of $g_{ab}(t)$ and that of $V_{ab}^b(t) = [\omega_{ab}^b(t), v_{ab}^b(t)]^T \in \mathfrak{se}(3)$, $\mathfrak{se}(3) = \mathfrak{so}(3) \times \mathbb{R}^3$ is the Lie algebra corresponding to SE(3) where $\mathfrak{so}(3)$ is the Lie algebra corresponding to SO(3) [6]. The right hand of (7) is the infinitesimal generator of the action corresponding to $V_{ab}^b(t)$,

and it is left invariant under left multiplication by constant matrices [27]. These properties determine the trivialization way of local coordinate map in Section 3.2.1.

2.2. Dynamics Model. The force equations of the aircraft are given by [28]

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} r \cdot v - q \cdot w \\ p \cdot w - r \cdot u \\ q \cdot u - p \cdot w \end{bmatrix} + \begin{bmatrix} -\sin \theta \\ \sin \phi \cdot \cos \theta \\ \cos \phi \cdot \cos \theta \end{bmatrix} g + \begin{bmatrix} \frac{\bar{q} \cdot s}{m} C_{X,t} \\ \frac{\bar{q} \cdot s}{m} C_{Y,t} \\ \frac{\bar{q} \cdot s}{m} C_{Z,t} \end{bmatrix} + \begin{bmatrix} \frac{T}{m} \\ 0 \\ 0 \end{bmatrix}, \quad (8)$$

where, \dot{u} , \dot{v} , and \dot{w} denote X -axis component, Y -axis component, and Z -axis component, respectively, of the aircraft

acceleration in the body-fixed frame, m is the aircraft mass, s is the wing reference area, \bar{q} is the free-stream dynamic pressure, T is the engine thrust, and $C_{X,t}$, $C_{Y,t}$, and $C_{Z,t}$ denote total X -axis force coefficient, total Y -axis force coefficient, and total Z -axis force coefficient, respectively.

One has

$$\begin{aligned} C_{X,t} &= C_X(\alpha, \beta) + \frac{\bar{c}q}{2V_T} C_{X_q}(\alpha), \\ C_{Y,t} &= C_Y(\alpha, \beta) + \frac{b}{2V_T} [C_{Y_r}(\alpha)r + C_{Y_p}(\alpha)p], \\ C_{Z,t} &= C_Z(\alpha, \beta) + \frac{\bar{c}q}{2V_T} C_{Z_q}(\alpha). \end{aligned} \quad (9)$$

Denote $[\bar{q}sC_{X,t}, \bar{q}sC_{Y,t}, \bar{q}sC_{Z,t}]^T$ and $[T, 0, 0]^T$ by $F_A^b(t)$ and $F_T^b(t)$, respectively; the former represents aerodynamics, and the latter represents thrust. Then, we have

$$m\dot{v}_{ab}^b(t) = -m\omega_{ab}^b(t) \times v_{ab}^b(t) + mgR_{ab}^T(t)e_3 + F_A^b(t) + F_T^b(t). \quad (10)$$

The moment equations of the aircraft are given by [26]

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{(J_{xz}(J_x - J_y + J_z)pq - (J_zJ_z - J_zJ_y + J_{xz}^2)rq + J_z\bar{q}sbC_{l,t} + J_{xz}\bar{q}sbC_{n,t})}{(J_xJ_z - J_{xz}^2)} \\ \frac{((J_z - J_x)pr - J_{xz}(p^2 - r^2) + \bar{q}s\bar{c}C_{m,t})}{J_y} \\ \frac{(((J_x - J_y)J_x + J_{xz}^2)pq - J_{xz}(J_x - J_y + J_z)rq + J_{xz}\bar{q}sbC_{l,t} + J_x\bar{q}sbC_{n,t})}{(J_xJ_z - J_{xz}^2)} \end{bmatrix}, \quad (11)$$

where, b is wing span, J_x , J_y , and J_z are moment of inertia about X -axis, Y -axis, and Z -axis, respectively, J_{xz} is cross-product of inertia, and $C_{l,t}$, $C_{m,t}$, and $C_{n,t}$ denote total rolling-moment coefficient, total pitching-moment coefficient, and total yawing-moment coefficient, respectively. The remaining parameters can refer to [26]

$$\begin{aligned} C_{l,t} &= C_l(\alpha, \beta) + \Delta C_{l,\delta_{a=20^\circ}} \frac{\delta_a}{20} + \Delta C_{l,\delta_{r=30^\circ}} \frac{\delta_r}{30} \\ &\quad + \frac{b}{2V_T} (C_{l_r}(\alpha)r + C_{l_p}(\alpha)p), \\ C_{m,t} &= C_m(\alpha, \beta)\eta_{\delta_h}(\delta_h) + C_{Z,t}(x_{cg,ref} - x_{cg}) + \frac{\bar{c}q}{2V_T} C_{m_q}(\alpha), \\ C_{n,t} &= C_n(\alpha, \beta) - C_{Y,t}(x_{cg,ref} - x_{cg}) \frac{\bar{c}}{b} + \Delta C_{n,\delta_{a=20^\circ}} \frac{\delta_a}{20} \\ &\quad + \Delta C_{n,\delta_{r=30^\circ}} \frac{\delta_r}{30} + \frac{b}{2V_T} (C_{n_r}(\alpha)r + C_{n_p}(\alpha)p). \end{aligned} \quad (12)$$

Denote $[\bar{q}sbC_{l,t}, \bar{q}s\bar{c}C_{m,t}, \bar{q}sbC_{n,t}]^T$ by $T_B(t)$, then we have

$$J\dot{\omega}_{ab}^b(t) = -\omega_{ab}^b(t) \times (J\omega_{ab}^b(t)) + T_B(t), \quad (13)$$

where $J = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix}$ is referred to as the inertia matrix of the rigid body.

Therefore, from the preceding equation and (10), we can construct a dynamics model $\dot{V}_{ab}^b(t) = \Pi^{-1} \cdot (B(V_{ab}^b(t)) \cdot V_{ab}^b(t) + F(t, \bar{g}_{ab}(t)))$ in $\mathfrak{se}(3)$, namely,

$$\begin{bmatrix} \dot{\omega}_{ab}^b(t) \\ \dot{v}_{ab}^b(t) \end{bmatrix} = \begin{bmatrix} J & 0 \\ 0 & mI_3 \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} \widehat{J\omega_{ab}^b(t)} & 0 \\ 0 & -m\widehat{\omega_{ab}^b(t)} \end{bmatrix} \begin{bmatrix} \omega_{ab}^b(t) \\ v_{ab}^b(t) \end{bmatrix} + \begin{bmatrix} T_B(t) \\ mgR_{ab}^T(t)e_3 + F_A^b(t) + F_T^b(t) \end{bmatrix} \right). \quad (14)$$

3. Geometric Pseudospectral Method on SE(3)

3.1. General Pseudospectral Method in Euclidean Space. General pseudospectral method will be used for computing the velocities in the Lie algebra. For this purpose, we briefly describe the basic principle of general pseudospectral method [24]. Let us take solving the following differential equation in \mathbb{R}^n at t_f , for example,

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^n, \quad t_0 \leq t \leq t_f. \quad (15)$$

Firstly, we equally divide a time interval $[t_0, t_f]$ into several time subintervals $[t_k, t_{k+1}]$ having a length h and transform the time interval $t \in [t_k, t_{k+1}]$ to the time interval $\tau \in [-1, 1]$ via the affine transformation

$$t = \frac{t_{k+1} - t_k}{2} \tau + \frac{t_{k+1} + t_k}{2}. \quad (16)$$

Thus, accordingly, (15) over $[t_k, t_{k+1}]$ becomes

$$\frac{dx(\tau)}{d\tau} = \frac{t_{k+1} - t_k}{2} f(\tau, x(\tau)), \quad -1 \leq \tau \leq 1. \quad (17)$$

Let $c_1 < \dots < c_N$ be collocation points in $[-1, 1]$ and $c_0 = -1$. We approximate the solution of (17) by the following formula:

$$x(\tau) \approx X(\tau) = \sum_{i=0}^N x_i \cdot L_i(\tau), \quad (18)$$

where $L_i(\tau) = \prod_{j=0, j \neq i}^N (\tau - c_j) / (c_i - c_j)$ is the Lagrange polynomials, satisfying the isolation property

$$L_i(c_j) = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \quad (19)$$

Equation (18) together with the isolation property leads to the fact that

$$x_i = x(c_i) \quad (20)$$

thus $X(c_j) = x(c_j)$.

Through expressing the derivative of the Lagrange polynomials at the collocation points in differential matrix form $D = [d_{ji}] \in \mathbb{R}^{N \times (N+1)}$,

$$d_{ji} = \dot{L}_i(c_j) = \sum_{l=0}^N \frac{\prod_{k=0, k \neq i, l}^N (c_j - c_k)}{\prod_{k=0, k \neq i}^N (c_i - c_k)}, \quad (21)$$

$$j = 1, \dots, N, \quad i = 0, \dots, N.$$

We can write (17) at the collocation points as a set of differential algebra equations as follows:

$$\sum_{i=0}^N d_{ji} \cdot X(c_i) = \frac{t_{k+1} - t_k}{2} f(c_j, X(c_j)), \quad j = 1, \dots, N. \quad (22)$$

Based on the previous equations, we establish defect equations

$$c_j = \sum_{i=0}^N d_{ji} \cdot X(c_i) - \frac{t_{k+1} - t_k}{2} f(c_j, X(c_j)), \quad j = 1, \dots, N \quad (23)$$

and apply iterative algorithms to the previous equations so that determining the approximation to $x(\tau)$ at the collocation points. Finally, according to the following formula, we obtain the approximation to (15) at the endpoint t_{k+1} of time subinterval $[t_k, t_{k+1}]$:

$$x(t_{k+1}) \approx X(t_k) + \frac{t_{k+1} - t_k}{2} \sum_{j=1}^N \omega_j \cdot f(c_j, X(c_j)), \quad (24)$$

where $\omega_j = \int_{-1}^1 L_j(t) dt$ is quadrature weight corresponding to the collocation points. This approximate solution will be initial value of $x(t)$ over $[t_{k+1}, t_{k+2}]$ and so on.

According to different selection methods of collocation points, pseudospectral methods can be divided into the standard method and the orthogonal method. Common collocation points in the orthogonal method are those obtained from the roots of either Chebyshev polynomials $T_N(t)$ or Legendre polynomials $P_N(t)$ belonging to the orthogonal polynomial [29]. The benefit of using the orthogonal method over the standard method is that the quadrature approximation to a definite integral is extremely accurate [24]. Furthermore, according to whether the endpoint is or is not a collocation point, pseudospectral methods fall into three general categories [30]: Gauss methods, neither of the endpoints -1 or 1 , is a collocation point; Radau methods, at most one of the endpoints -1 or 1 , are a collocation point; Lobatto methods, both of the endpoints -1 and 1 , are collocation points. In the rest of this paper, we will develop our geometric pseudospectral method based on Legendre-Gauss points, abbreviated as Gauss points in the following, for two main reasons: firstly, there is an intimate relationship between Gauss method and some implicit RK method whose coefficients in the Butcher Tableau can be used by our method for computing configuration [2]; secondly, when Gauss method is used for transcribing the continues optimal control problem into a discrete nonlinear programming problem, it does not suffer from a defect in the optimality conditions at the boundary points due to the endpoints being not collocation points [23].

3.2. Geometric Pseudospectral Method on SE(3). First recall and rewrite the aircraft rigid-body dynamics model Equations (7) and (14) in Section 2 as follows:

$$\dot{\bar{g}}_{ab}(t) = \bar{g}_{ab}(t) \cdot \bar{V}_{ab}^b(t), \quad (25)$$

$$\dot{V}_{ab}^b(t) = \Pi^{-1} \cdot (B(V_{ab}^b(t)) \cdot V_{ab}^b(t) + F(t, \bar{g}_{ab}(t))) \quad (26)$$

$$\triangleq f(V_{ab}^b(t), \bar{g}_{ab}(t)).$$

Step 1. Initialization:

Initialize time interval $[t_0, t_f]$, step length h , configuration $\bar{g}_{ab}(t_0)$, velocity $V_{ab}^b(t_0)$, and let k be 0;

Step 2. Main loop:

Step 2.1. Compute the Gauss points $\{\tau_1, \tau_2\}$ in $[t_k, t_{k+1}]$, corresponding differentiation matrix $D_{2 \times 3}$ and quadrature weights $\{\omega_1, \omega_2\}$;

Step 2.2. Compute velocities $V_{ab}^b(\tau_1)$ and $V_{ab}^b(\tau_2)$ at the Gauss points:

Step 2.2.1. Set the number of iterations, threshold of iterations deviation and let current step be 1;

Step 2.2.2. Let initial value of both $\bar{g}_{ab}^{(step)}(\tau_1)$ and $\bar{g}_{ab}^{(step)}(\tau_2)$ be $\bar{g}_{ab}(t_k)$, that of both $V_{ab}^{b(step)}(\tau_1)$ and $V_{ab}^{b(step)}(\tau_2)$ be $V_{ab}^b(t_k)$;

Step 2.2.3. Child loop:

Step 2.2.3.1. According to the following equations, obtain $[\dot{V}_{ab}^{b(step)}(\tau_1), \dot{V}_{ab}^{b(step)}(\tau_2)]^T$,

$$\begin{bmatrix} \dot{V}_{ab}^{b(step)}(\tau_1) \\ \dot{V}_{ab}^{b(step)}(\tau_2) \end{bmatrix} = \frac{t_{k+1} - t_k}{2} \begin{bmatrix} \Pi^{-1} \cdot (B(V_{ab}^{b(step)}(\tau_1)) \cdot V_{ab}^{b(step)}(\tau_1) + F(\tau_1, \bar{g}_{ab}^{(step)}(\tau_1))) \\ \Pi^{-1} \cdot (B(V_{ab}^{b(step)}(\tau_2)) \cdot V_{ab}^{b(step)}(\tau_2) + F(\tau_2, \bar{g}_{ab}^{(step)}(\tau_2))) \end{bmatrix} - \begin{bmatrix} d_{01} \\ d_{02} \end{bmatrix} V_{ab}^b(t_k)$$

Step 2.2.3.2. Update velocities $[V_{ab}^{b(step+1)}(\tau_1), V_{ab}^{b(step+1)}(\tau_2)]^T$ at the Gauss point,

$$\begin{bmatrix} V_{ab}^{b(step+1)}(\tau_1) \\ V_{ab}^{b(step+1)}(\tau_2) \end{bmatrix} = \bar{D}^{-1} \begin{bmatrix} \dot{V}_{ab}^{b(step)}(\tau_1) \\ \dot{V}_{ab}^{b(step)}(\tau_2) \end{bmatrix}$$

Step 2.2.3.3. Compute 4th-order reduced truncated Magnus series expansion $[u^{[4]}(\tau_1), u^{[4]}(\tau_2)]^T$ at the Gauss points,

$$\begin{bmatrix} u^{[4]}(\tau_1) \\ u^{[4]}(\tau_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{4}h \cdot V_{ab}^{b(step)}(\tau_1) + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)h \cdot V_{ab}^{b(step)}(\tau_2) - \frac{1}{2} \left(\frac{5}{72} - \frac{\sqrt{3}}{24}\right)h^2 \cdot [V_{ab}^{b(step)}(\tau_1), V_{ab}^{b(step)}(\tau_2)] \\ \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)h \cdot V_{ab}^{b(step)}(\tau_1) + \frac{1}{4}h \cdot V_{ab}^{b(step)}(\tau_2) - \frac{1}{2} \left(-\frac{5}{72} - \frac{\sqrt{3}}{24}\right)h^2 \cdot [V_{ab}^{b(step)}(\tau_1), V_{ab}^{b(step)}(\tau_2)] \end{bmatrix}$$

Step 2.2.3.4. Using the Cayley map, update configurations $[\bar{g}_{ab}^{(step+1)}(\tau_1), \bar{g}_{ab}^{(step+1)}(\tau_2)]^T$ at the Gauss points,

$$\begin{bmatrix} \bar{g}_{ab}^{(step+1)}(\tau_1) \\ \bar{g}_{ab}^{(step+1)}(\tau_2) \end{bmatrix} = \begin{bmatrix} \bar{g}_{ab}(t_k) \cdot \text{cay}(u^{[4]}(\tau_1)) \\ \bar{g}_{ab}(t_k) \cdot \text{cay}(u^{[4]}(\tau_2)) \end{bmatrix}$$

Step 2.2.3.5. Compute the configuration deviation and velocity deviation respectively between the adjacent steps, and let the larger one between them be iterative deviation at the current step,

$$\text{norm}(\bar{g}_{ab}^{(err)}) = \left\| \begin{bmatrix} \log_{\text{SE}(3)}(\bar{g}_{ab}^{(step+1)}(\tau_1)^{-1} \cdot \bar{g}_{ab}^{(step)}(\tau_1)) \\ \log_{\text{SE}(3)}(\bar{g}_{ab}^{(step+1)}(\tau_2)^{-1} \cdot \bar{g}_{ab}^{(step)}(\tau_2)) \end{bmatrix} \right\|_2, \text{norm}(V_{ab}^{b(err)}) = \left\| \begin{bmatrix} \|V_{ab}^{b(step+1)}(\tau_1) - V_{ab}^{b(step)}(\tau_1)\|_2 \\ \|V_{ab}^{b(step+1)}(\tau_2) - V_{ab}^{b(step)}(\tau_2)\|_2 \end{bmatrix} \right\|_\infty$$

Step 2.2.3.6. Substitute $\bar{g}_{ab}^{(step+1)}$ and $V_{ab}^{b(step+1)}$ at τ_1 and τ_2 into $\bar{g}_{ab}^{(step)}$ and $V_{ab}^{b(step)}$ respectively, at the same time, let step plus 1;

Step 2.2.3.7. Compute the difference between iterative deviation and specified tolerance, so that determine whether or not to terminate the iterative process.

Step 2.2.4. End child loop;

Step 2.2.5. Use the final iterative results $\bar{g}_{ab}^{(step)}$ and $V_{ab}^{b(step)}$ in child loop to compute the velocity $V_{ab}^b(t_{k+1})$ at the endpoint t_{k+1} ,

$$V_{ab}^b(t_{k+1}) = V_{ab}^b(t_k) + \frac{t_{k+1} - t_k}{2} \sum_{i=1}^2 \omega_i \cdot \Pi^{-1} \cdot (B(V_{ab}^{b(step)}(\tau_i)) V_{ab}^{b(step)}(\tau_i) + F(\tau_i, \bar{g}_{ab}^{(step)}(\tau_i)))$$

Step 2.2.6. Compute configuration $\bar{g}_{ab}(t_{k+1})$ at t_{k+1} :

Step 2.2.6.1. Compute 4th-order reduced truncated Magnus series expansion $u^{[4]}(t_{k+1})$ at t_{k+1} ,

$$u^{[4]}(t_{k+1}) = \frac{1}{2}h \cdot (V_{ab}^{b(step)}(\tau_1) + V_{ab}^{b(step)}(\tau_2)) + \frac{\sqrt{3}}{12}h^2 \cdot [V_{ab}^{b(step)}(\tau_1), V_{ab}^{b(step)}(\tau_2)]$$

Step 2.2.6.2. Use the Cayley map to update configuration $\bar{g}_{ab}(t_{k+1})$,

$$\bar{g}_{ab}(t_{k+1}) = \bar{g}_{ab}(t_k) \cdot \text{cay}(u^{[4]}(t_{k+1}))$$

Step 2.2.7. Compare t_{k+1} with t_f , so that determine whether or not to terminate the main loop;

Step 3. End loop.

ALGORITHM 1: 4th-order geometric pseudospectral algorithm on SE(3).

For the kinematics Equation (25), it is well known that the solution of (25) stays on SE(3) for all $t \geq t_0$. How to use pseudospectral method to solve (25) while preserving the structural feature of the differential equation under discretization of $\bar{g}_{ab}(t)$ is the main problem to be solved in this section. In \mathbb{R}^n , both the solution space and the tangent space are linear vector space. Classical numerical methods, including general pseudospectral method, just rely on domain space consisting of vectors. However, the special

Euclidean group is a nonlinear manifold, so using classical numerical methods directly to solve differential equation on SE(3) will be not able to preserve its Lie group structure. Reference [21] indicated that any differential equation in the form of an infinitesimal generator on a homogeneous space (a manifold with a transitive Lie group action) is shown to be locally equivalent to a differential equation on the Lie algebra corresponding to the Lie group acting on the homogenous space. Also, a Lie algebra corresponding to the Lie group is a

vector space with the additional structure of a commutator. For the previous reasons, the Lie algebra is the natural choice of space for our geometric pseudospectral method. First, we will apply the equivariant map to transform (25) evolving on SE(3) into an equivalent differential equation evolving in $\mathfrak{se}(3)$. Next, we use the Gauss pseudospectral method to solve the equivalent differential equation in order to obtain p th-order truncated approximation of the equation at the Gauss points and the endpoints, followed by mapping the approximate solution back to configuration space via local coordinate map f .

For the dynamics equation (26), because the velocity $V_{ab}^b(t)$ belongs to $\mathfrak{se}(3)$, and actually $\mathfrak{se}(3)$ is isomorphic to the \mathbb{R}^6 , we can use Gauss pseudospectral method directly on (26) to compute the velocities at the Gauss points and the endpoint.

3.2.1. Computing Configurations at the Gauss Points and the Endpoint. In order to transform (25) on SE(3) into an equivalent differential equation evolving on $\mathfrak{se}(3)$, we briefly describe the basic idea of equivariant map.

Definition 1 (equivariant map). Let \mathcal{M} and \mathcal{N} be manifolds, and let G be a Lie group which acts on \mathcal{M} by $\Phi_g : \mathcal{M} \rightarrow \mathcal{M}$ and on \mathcal{N} by $\Psi_g : \mathcal{N} \rightarrow \mathcal{N}$. A smooth map $f : \mathcal{M} \rightarrow \mathcal{N}$ is called equivariant with respect to these actions if, for all $g \in G$,

$$f \circ \Phi_g = \Psi_g \circ f \quad (27)$$

that is, the following diagram commutes of equivariant map f :

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{f} & \mathcal{N} \\ \Phi_g \uparrow & (27) & \uparrow \Psi_g \\ \mathcal{M} & \xrightarrow{f} & \mathcal{N} \end{array} \quad (28)$$

First, from the definition of an action of G on \mathcal{M} , we can obtain an equivariant map $\Phi_y : G \rightarrow \mathcal{M}$ with respect to the left translation action L_g of G on itself and an action G of Φ_g on \mathcal{M} as follows:

$$\Phi_y \circ L_g = \Phi_g \circ \Phi_y. \quad (29)$$

It is known that there is a local coordinate map $f : \mathfrak{g} \rightarrow G$ on G with the Lie algebra \mathfrak{g} ; the most typical one is exponential map $\exp : \mathfrak{g} \rightarrow G$. At this point, we need to find an action B_g of G on \mathfrak{g} such that f will be an equivariant map with B_g and the left action of G on itself, namely,

$$f \circ B_g = L_g \circ f. \quad (30)$$

In the case where f is the exponential map, B_g is nothing else than the well-known Baker-Campbell-Hausdorff (BCH) formula:

$$B_g(u) = \log(g \cdot \exp(u)), \quad (31)$$

where $\log : G \rightarrow \mathfrak{g}$ is called the logarithm map. Since composition of two equivariant maps is also an equivariant map, we can construct an equivariant map $\Phi_y \circ f$ from \mathfrak{g} to \mathcal{M} with respect to the action B_g on \mathfrak{g} and Φ on \mathcal{M} . Diagram commutes of composition $\Phi_y \circ f$ are as follows:

$$\begin{array}{ccccc} & & f & & \Phi_y \\ & & \rightarrow & & \rightarrow \\ \mathfrak{g} & & & G & & \mathcal{M} \\ & \uparrow & (30) & \uparrow & (29) & \uparrow \\ B_{\exp(t\xi)} & & & L_g & & \Phi_{\exp(t\xi)} \\ & & f & & \Phi_y \\ \mathfrak{g} & & \rightarrow & G & & \mathcal{M} \end{array} \quad (32)$$

The theorem 3.6 of [21] stated that if ϕ is an equivariant map, then the infinitesimal generators of the action with respect to the same element $\xi \in \mathfrak{g}$ are ϕ -related vector fields. Thus, the infinitesimal generators $\xi_{\mathfrak{g}}$ and $\xi_{\mathcal{M}}$ of the flows $B_{\exp(t\xi)}$ and $\Phi_{\exp(t\xi)}$, respectively, on \mathfrak{g} and \mathcal{M} are $\Phi_y \circ f$ -related; that is,

$$\xi_{\mathcal{M}} \circ \Phi_y \circ f = T\Phi_y \circ Tf \circ \xi_{\mathfrak{g}}. \quad (33)$$

Finally, we need to determine what $\xi_{\mathfrak{g}}$ is, and the following theorem gives the conditions that it needs to meet.

Theorem 2 (see [21]). Let $f : \mathfrak{g} \rightarrow G$ be a coordinate map on G and $\Phi_y \circ f$ equivariant with respect to the flows $B_{\exp(t\xi)}$ and $\Phi_{\exp(t\xi)}$. The infinitesimal generator of B_g satisfying (33) is $\xi_{\mathfrak{g}}(u) = df_u^{-1}(v)$. $df : \mathfrak{g} \rightarrow \mathfrak{g}$ is the trivialization of Tf , defined as $df_u = TR_{f(u)^{-1}} \circ Tf_u$.

The following commutative diagram describing the equivariant maps of composition $\Phi_y \circ f$ (left) and its infinitesimal generator (right) sums up the previous processes:

$$\begin{array}{ccccc} & \Phi_y \circ f & & T\Phi_y \circ Tf & \\ \mathfrak{g} & \rightarrow & \mathcal{M} & \rightarrow & T\mathcal{M} \\ B_g \uparrow & & \uparrow \Phi_g & \Rightarrow & \uparrow \xi_{\mathcal{M}}(y) \\ \mathfrak{g} & \xrightarrow{\Phi_y \circ f} & \mathcal{M} & & \mathfrak{g} \xrightarrow{\Phi_y \circ f} \mathcal{M} \\ & & & df_u^{-1}(v) \uparrow & (33) \end{array} \quad (34)$$

Also, [21] stated that any differential equation on \mathcal{M} which can be written as an infinitesimal generator fits into the previous framework, including all Lie type equations, isospectral flows, and rigid frames. It is noted that, as mentioned in Section 2.1, the right hand of (7) or (25) is an infinitesimal generator on SE(3), and here we assume that the solution $\bar{g}_{ab}(t)$ of (25) can be written as the following form [13]:

$$\bar{g}_{ab}(t) = \bar{g}_{ab}(t_0) \cdot f(u(t)). \quad (35)$$

In order to obtain the explicit expression of $u(t)$, we need to solve the following differential equation:

$$\dot{u}(t) = df_{-u}^{-1}(V_{ab}^b(t)). \quad (36)$$

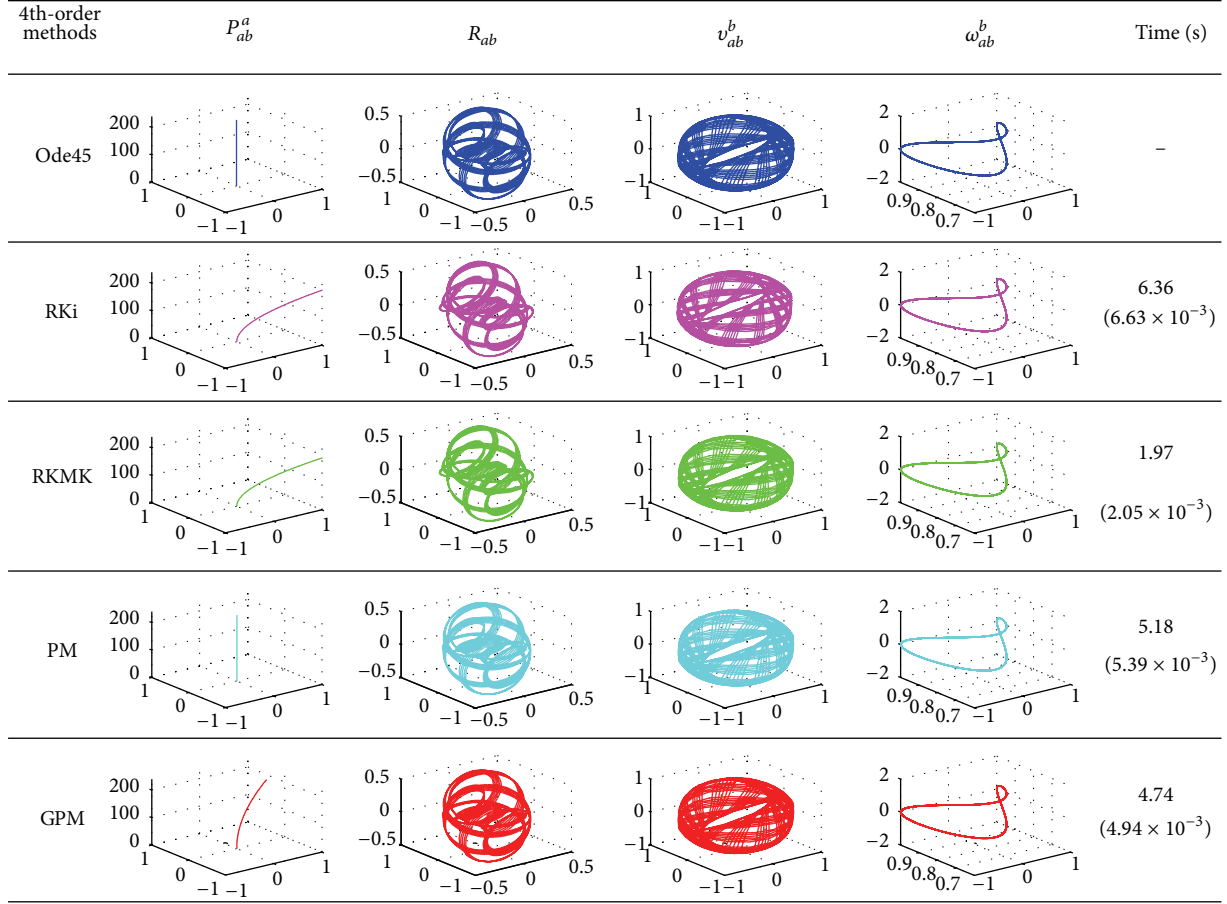


FIGURE 1: Position (P_{ab}^a), orientation (R_{ab}), linear velocity (v_{ab}^b), and angular velocity (ω_{ab}^b) versus time and total computational time with average time per step for the free-floating rigid body when step size is equal to 0.25 s.

For deriving the previous formula, we differentiate (35) as follows:

$$\begin{aligned}
 \frac{d}{dt} \bar{g}_{ab}(t) &= \frac{d}{dt} (\bar{g}_{ab}(t_0) \cdot f(u(t))) \\
 &= \bar{g}_{ab}(t_0) \cdot dL_{f(u(t))} \circ df_{-u(t)}(\dot{u}(t)) \\
 &= \bar{g}_{ab}(t_0) \cdot f(u(t)) \cdot df_{-u(t)}(\dot{u}(t)).
 \end{aligned} \tag{37}$$

Comparing (37) with (25) and (35), we have

$$V_{ab}^b(t) = df_{-u(t)}(\dot{u}(t)). \tag{38}$$

Taking the inverse of previous formula, we obtain (36).

It seems from the equivariant map and the previous formula derivation that solving equation (25) can be transformed into a research on the equivalent equation (36), and the manifold \mathcal{M} is SE(3) at this point. $df_{-u}^{-1}(V_{ab}^b(t))$ has different forms, according to the different choices of local coordinate map f . In the case where f is the exponential map, the vector field df_{-u}^{-1} on \mathfrak{g} can be represented as [13]

$$\begin{aligned}
 d \exp_{-u}^{-1}(V_{ab}^b(t)) &= V_{ab}^b(t) + \frac{1}{2} \text{ad}_u(V_{ab}^b(t)) \\
 &\quad + \sum_{k=2}^{\infty} \frac{B_k}{k!} \text{ad}_u^k(V_{ab}^b(t)),
 \end{aligned} \tag{39}$$

where the adjoint operator $\text{ad}_u : \mathfrak{g} \rightarrow \mathfrak{g}$ is Lie bracket or commutator, $\text{ad}_u(V_{ab}^b(t)) = [u, V_{ab}^b(t)]$, satisfying recursive expression $\text{ad}_u^k(V_{ab}^b(t)) = [u, \text{ad}_u^{k-1}(V_{ab}^b(t))]$, and B_k are Bernoulli numbers [31] as follows:

$$B_k = \begin{cases} \frac{1}{6}, -\frac{1}{30}, \frac{1}{42}, -\frac{1}{30}, \frac{5}{66}, -\frac{691}{2730}, \dots, & k = 2N, N \in \mathbb{Z}^+, \\ 0, & k = 2N + 1, N \in \mathbb{Z}^+. \end{cases} \tag{40}$$

In the context of rigid-body motion, the right trivialization corresponds to the differential equation with tangent vectors $AY \in \mathfrak{X}(\mathcal{M})$, $A \in \mathfrak{g}$, and $Y \in G$, and the left trivialization corresponds to the differential equation with tangent vectors $YA \in \mathfrak{X}(\mathcal{M})$, $A \in \mathfrak{g}$, and $Y \in G$. The former represents the change of configuration in a space frame, and the latter represents the change of configuration in the body-fixed frame. Here, let df be the left trivialization of Tf , since the configuration model established in Section 2.1 is in the body-fixed frame. In the case of df being the left trivialization and f being the exponential map, we can find an explicit

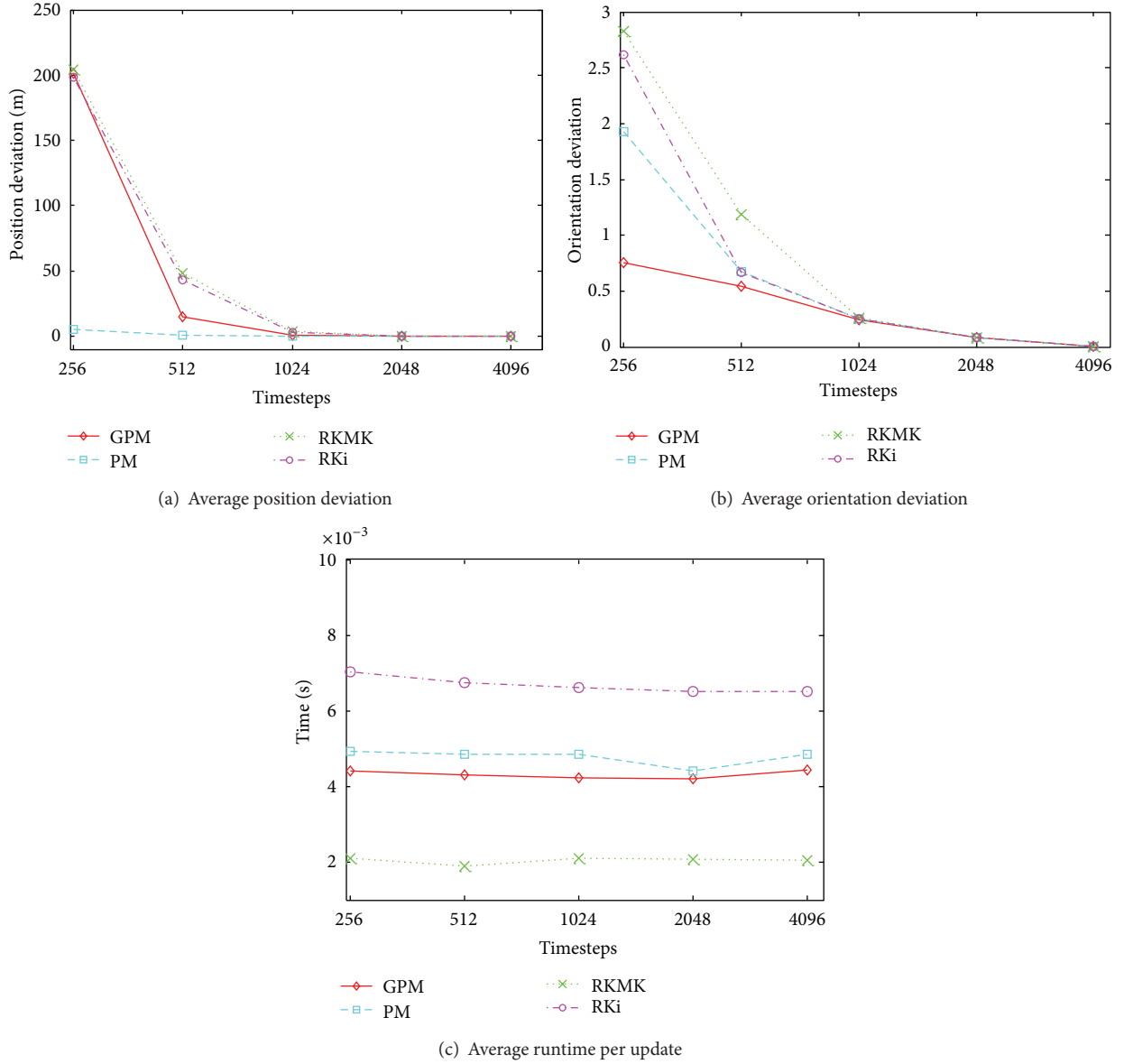


FIGURE 2: Accuracy and computational efficiency of GPM compared with PM, RKMK, and RKi.

approximation to $u(t)$ called the Magnus series expansion via applying Picard iterations to (36) [31] as follows:

$$\begin{aligned}
 u(t) = & \int_0^t V_{ab}^b(\xi_1) d\xi_1 + \frac{1}{2} \int_0^t \left[\int_0^{\xi_1} V_{ab}^b(\xi_2) d\xi_2, V_{ab}^b(\xi_1) \right] d\xi_1 \\
 & + \frac{1}{12} \int_0^t \left[\int_0^{\xi_1} V_{ab}^b(\xi_2) d\xi_2, \left[\int_0^{\xi_1} V_{ab}^b(\xi_2) d\xi_2, V_{ab}^b(\xi_1) \right] \right] \\
 & \times d\xi_1 + \dots
 \end{aligned} \tag{41}$$

The previous formula can be rewritten as an infinite sum $u(t) = \sum_{k=0}^{\infty} H_k(t)$, where each $H_k(t)$ is a linear combination of terms that have equal numbers k of integrals (exclude the

outermost integral from 0 to t) and commutators. We denote each term of $H_k(t)$ by $C_\kappa(\xi)$ and associate it with a binary tree $\kappa \in \mathbb{T}_k$, where \mathbb{T}_k is a set which includes all trees with the same number k of integrals and commutators. Then, we can rewrite (41) as

$$u(t) = \sum_{k=0}^{\infty} \sum_{\kappa \in \mathbb{T}_k} \alpha(\kappa) \int_0^t C_\kappa(\xi) d\xi, \quad t \geq 0, \tag{42}$$

where

$$\alpha(\kappa) = \frac{\tilde{B}_k}{k!} \prod_{i=1}^k \alpha(\kappa_i), \quad k \in \mathbb{Z}^+. \tag{43}$$

TABLE I: Parameters of numerical test.

Time	$t \in [0, 240]$ s
Mass	$m = 1$ kg
Moment of inertia	$J = \text{diag}[1, 2.8, 2]$ kg·m ²
Initial orientation	$R_{ab}(0) = I_{3 \times 3}$
Initial position	$p_{ab}^a(0) = [0, 0, 0]^T$ m
Initial linear velocity	$v_{ab}^b(0) = [0, 0, 1]^T$ m/s
Initial angular velocity	$\omega_{ab}^b(0) = [1, 1, 0]^T$ rad/s

It is worthwhile to note that, unlike right trivialization in [13], here, we take left-trivialized tangent of local coordinate map, thus

$$\tilde{B}_k = \begin{cases} \frac{1}{2}, & \text{for } k = 1, \\ B_k, & \text{for } k \neq 1. \end{cases} \quad (44)$$

Based on the binary tree theory, we can find the following p th-order approximate solution of (36),

$$u^{[p]}(t) = \sum_{m=0}^{p-1} \sum_{\kappa \in \mathbb{F}_m} \alpha(\kappa) \int_0^t C_\kappa(\xi) d\xi, \quad (45)$$

where \mathbb{F}_m is the set of all m -power rooted trees. Thereby, we give the Magnus series terms with its coefficients $\alpha(\kappa)$ under left trivialization in Table 2, which correspond to the top eight orders rooted trees.

Then, we use Gauss pseudospectral method to solve (45), so that we get the approximate solution of (36) at the Gauss points $t_k < \tau_1, \dots, \tau_N < t_{k+1}$ and the endpoint $t_{k+1} \triangleq t_k + h$. To begin with, assume that all of velocities $V_{ab}^b(\tau_i)$ at the Gauss points are known previously; we use the Lagrange polynomials at these points to approximate the velocity at any time $t \in [t_k, t_{k+1}]$ as follows:

$$V_{ab}^b(t) \approx \sum_{i=1}^N L_i \left(\frac{t - t_k}{h} \right) \cdot V_{ab}^b(\tau_i). \quad (46)$$

Next, through transforming (46) into $[0, 1]$ via an affine transformation $\xi = (t - t_k)/h$, and inserting it to (45), we have

$$\begin{aligned} u^{[p]}(t) &= I_1(t) + \frac{1}{2}I_2(t) + \frac{1}{4}I_3(t) + \frac{1}{12}I_4(t) + \frac{1}{24}I_5(t) \\ &+ \frac{1}{24}I_6(t) + \frac{1}{8}I_8(t) + \dots, \end{aligned} \quad (47)$$

where each term $I(t)$ is the following form of multivariate quadrature:

$$\begin{aligned} I(t) &= \int_0^t \int_0^{\xi_{m_2}} \int_0^{\xi_{m_3}} \dots \end{aligned}$$

$$\begin{aligned} &\int_0^{\xi_{m_s}} \mathcal{L}(h \cdot V_{ab}^b(\xi_1), \dots, h \cdot V_{ab}^b(\xi_s)) d\xi_s \dots d\xi_2 d\xi_1 \\ &= \int_S \mathcal{L}(h \cdot V_{ab}^b(\xi_1), \dots, h \cdot V_{ab}^b(\xi_s)) d\xi_s \dots d\xi_2 d\xi_1, \end{aligned} \quad (48)$$

where $S = \{\xi_1, \dots, \xi_s \in \mathbb{R} | \xi_1 \in [0, t], \xi_\ell \in [0, \xi_{m_\ell}], m_\ell \in \{1, 2, \dots, \ell - 1\}, \ell = 2, 3, \dots, s\}$ and \mathcal{L} denotes multivariate function.

The same as the univariate case, we use the following quadrature formulae (49) and (50) to evaluate the approximate solution of (48) at τ_1, \dots, τ_N and t_{k+1} , respectively:

$$\begin{aligned} I(\tau_i) &= \int_{\tilde{S}^i} \mathcal{L}(h \cdot V_{ab}^b(\xi_1), \dots, h \cdot V_{ab}^b(\xi_{s-1})) d\xi_{s-1} \dots d\xi_2 d\xi_1 \\ &\approx \sum_{k \in C_{s-1}^N} a_k^i \mathcal{L}(h \cdot V_{ab}^b(\xi_{k_1}), \dots, h \cdot V_{ab}^b(\xi_{k_{s-1}})), \\ a_k^i &= \int_{\tilde{S}^i} \prod_{\ell=1}^{s-1} L_{k_\ell}(\xi_\ell) d\xi_\ell, \quad k_1, k_2, \dots, k_{s-1} \in k, \end{aligned} \quad (49)$$

where $\tilde{S}^i = \{\xi_1, \dots, \xi_{s-1} \in \mathbb{R} | \xi_1 \in [0, c_i], \xi_\ell \in [0, \xi_{m_\ell}], m_\ell \in \{1, 2, \dots, \ell - 1\}, \ell = 2, 3, \dots, s - 1\}$, and C_{s-1}^N is the set of all combinations of $(s - 1)$ -tuples k from the set $\{1, 2, \dots, N\}$, excluding $s - 1$ repeats of the same number in the set.

One has

$$\begin{aligned} I(t_{k+1}) &= \int_{\tilde{S}} \mathcal{L}(h \cdot V_{ab}^b(\xi_1), \dots, h \cdot V_{ab}^b(\xi_s)) d\xi_s \dots d\xi_2 d\xi_1 \\ &\approx \sum_{k \in C_s^N} b_k \mathcal{L}(h \cdot V_{ab}^b(\xi_{k_1}), \dots, h \cdot V_{ab}^b(\xi_{k_s})) \\ b_k &= \int_{\tilde{S}} \prod_{\ell=1}^s L_{k_\ell}(\xi_\ell) d\xi_\ell, \quad k_1, k_2, \dots, k_s \in k, \end{aligned} \quad (50)$$

where $\tilde{S} = \{\xi_1, \dots, \xi_s \in \mathbb{R} | \xi_1 \in [0, 1], \xi_\ell \in [0, \xi_{m_\ell}], m_\ell \in \{1, 2, \dots, \ell - 1\}, \ell = 2, 3, \dots, s\}$ and C_s^N is similar to C_{s-1}^N .

Through computing $I(t)$ of (47) term by term, we obtain p th-order approximate solution $u^{[p]}(t)$ of $u(t)$ at the discrete points. As seen from (48), $I(t)$ depends on the velocities at the Gauss points, and the velocities need to be solved in an iterative manner which will be explained in next section. Thereby, the solutions of configuration at the Gauss points need to be also determined in an iterative manner. Finally, we use $u^{[p]}(\tau_i)$ at the Gauss points to compute configuration corresponding to the same points via the following formula:

$$\bar{g}_{ab}(\tau_i) = \bar{g}_{ab}(t_k) \cdot f(u^{[p]}(\tau_i)). \quad (51)$$

There are several choices of local coordinate map $f: \mathfrak{g} \rightarrow G$, such as exponential map and Cayley map [21]. We incline

TABLE 2: Magnus series terms corresponding to the top eight orders rooted trees.

p	\mathbb{T}_k	\mathbb{F}_m	$\int_0^t C_\tau(\xi_1) d\xi_1$	$I(t)$	$\alpha(\kappa)$
1	\mathbb{T}_0	\mathbb{F}_0	$\int_0^t V_{ab}^b(\xi_1) d\xi_1$	$I_1(t)$	1
3	\mathbb{T}_1	\mathbb{F}_2	$\int_0^t \int_0^{\xi_1} [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)] d\xi_2 d\xi_1$	$I_2(t)$	1/2
4	\mathbb{T}_2	\mathbb{F}_3	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)] d\xi_3 d\xi_2 d\xi_1$	$I_3(t)$	1/4
4	\mathbb{T}_2	\mathbb{F}_3	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_1} [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]] d\xi_2 d\xi_2 d\xi_1$	$I_4(t)$	1/12
5	\mathbb{T}_3	\mathbb{F}_4	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_2} [[V_{ab}^b(\xi_3), [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)]]], V_{ab}^b(\xi_1)] d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_5(t)$	1/24
5	\mathbb{T}_3	\mathbb{F}_4	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_1} \int_0^{\xi_2} [V_{ab}^b(\xi_2), [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)]] d\xi_3 d\xi_2 d\xi_2 d\xi_1$	$I_6(t)$	1/24
7	\mathbb{T}_3	\mathbb{F}_6	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_1} [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]] d\xi_3 d\xi_2 d\xi_2 d\xi_1$	$I_7(t)$	1/24
5	\mathbb{T}_3	\mathbb{F}_4	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [[V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)], V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)] d\xi_4 d\xi_3 d\xi_2 d\xi_1$	$I_8(t)$	1/8
5	\mathbb{T}_3	\mathbb{F}_4	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_1} \int_0^{\xi_1} [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]]] d\xi_2 d\xi_2 d\xi_2 d\xi_1$	$I_9(t)$	0
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} \int_0^{\xi_4} [[[[V_{ab}^b(\xi_5), V_{ab}^b(\xi_4)], V_{ab}^b(\xi_3)], V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)] d\xi_5 d\xi_4 d\xi_3 d\xi_2 d\xi_1$	$I_{10}(t)$	1/16
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [[V_{ab}^b(\xi_4), [V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)]]], V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)] d\xi_4 d\xi_4 d\xi_3 d\xi_2 d\xi_1$	$I_{11}(t)$	1/48
8	\mathbb{T}_4	\mathbb{F}_7	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} \int_0^{\xi_4} [[V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)], [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)]]], V_{ab}^b(\xi_1)] d\xi_4 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{12}(t)$	1/48
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [[V_{ab}^b(\xi_3), [V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)], V_{ab}^b(\xi_2)]]], V_{ab}^b(\xi_1)] d\xi_4 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{13}(t)$	1/48
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_2} [[V_{ab}^b(\xi_3), [V_{ab}^b(\xi_3), [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)]]]]], V_{ab}^b(\xi_1)] d\xi_3 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{14}(t)$	0
7	\mathbb{T}_4	\mathbb{F}_6	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [[V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)], V_{ab}^b(\xi_2)], [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]] d\xi_4 d\xi_3 d\xi_2 d\xi_2 d\xi_1$	$I_{15}(t)$	1/48
7	\mathbb{T}_4	\mathbb{F}_6	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_2} [[V_{ab}^b(\xi_3), [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)]]], [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]] d\xi_3 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{16}(t)$	1/144
7	\mathbb{T}_4	\mathbb{F}_6	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_2} [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)]] d\xi_3 d\xi_3 d\xi_2 d\xi_2 d\xi_1$	$I_{17}(t)$	1/48
7	\mathbb{T}_4	\mathbb{F}_6	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]]]] d\xi_3 d\xi_2 d\xi_2 d\xi_2 d\xi_1$	$I_{18}(t)$	0
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [V_{ab}^b(\xi_2), [[V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)], V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)]] d\xi_4 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{19}(t)$	1/48
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [V_{ab}^b(\xi_2), [[V_{ab}^b(\xi_4), [V_{ab}^b(\xi_4), V_{ab}^b(\xi_3)]]], V_{ab}^b(\xi_1)]] d\xi_4 d\xi_4 d\xi_3 d\xi_2 d\xi_1$	$I_{20}(t)$	1/144
8	\mathbb{T}_4	\mathbb{F}_7	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_3} [V_{ab}^b(\xi_2), [[V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]]]] d\xi_3 d\xi_2 d\xi_2 d\xi_2 d\xi_1$	$I_{21}(t)$	0
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_2} [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_3), V_{ab}^b(\xi_2)], V_{ab}^b(\xi_1)]]] d\xi_3 d\xi_3 d\xi_3 d\xi_2 d\xi_1$	$I_{22}(t)$	0
6	\mathbb{T}_4	\mathbb{F}_5	$\int_0^t \int_0^{\xi_1} \int_0^{\xi_2} \int_0^{\xi_1} [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), [V_{ab}^b(\xi_2), V_{ab}^b(\xi_1)]]]]] d\xi_2 d\xi_2 d\xi_2 d\xi_2 d\xi_1$	$I_{23}(t)$	-1/720

TABLE 3: The top eight orders reduced truncated Magnus series expansion.

Order	q	p	Reduced Magnus series expansion
2	1	1	$u^{[2]}(t) = I_1(t)$
4	2	3	$u^{[4]}(t) = I_1(t) + \frac{1}{2}I_2(t)$
6	3	5	$u^{[6]}(t) = I_1(t) + \frac{1}{2}I_2(t) + \frac{1}{4}I_3(t) + \frac{1}{12}I_4(t) + \frac{1}{24}I_5(t) + \frac{1}{24}I_6(t) + \frac{1}{8}I_8(t)$
8	4	7	$u^{[8]}(t) = I_1(t) + \frac{1}{2}I_2(t) + \frac{1}{4}I_3(t) + \frac{1}{12}I_4(t) + \frac{1}{24}I_5(t) + \frac{1}{24}I_6(t) + \frac{1}{8}I_8(t) + \frac{1}{16}I_{10}(t) + \frac{1}{48}I_{11}(t) + \frac{1}{48}I_{13}(t) + \frac{1}{48}I_{15}(t) + \frac{1}{48}I_{19}(t) + \frac{1}{144}I_{20}(t) - \frac{1}{720}I_{23}(t) + \frac{1}{24}I_7(t) + \frac{1}{48}I_{15}(t) + \frac{1}{144}I_{16}(t) + \frac{1}{48}I_{17}(t)$

to the Cayley map, since it is very easy to compute, despite it is only an approximation to the integral curve defined by exponential map as follows:

$$\text{cay}_{\text{SE}(3)}(u) = \begin{bmatrix} \text{cay}_{\text{SO}(3)}(\omega) & A_{\text{cay}}(\omega) \cdot v \\ 0 & 1 \end{bmatrix} \quad (52)$$

$$\text{cay}_{\text{SO}(3)}(\omega) = I_3 + \frac{4}{4 + \|\omega\|^2} \left(\hat{\omega} + \frac{(\hat{\omega})^2}{2} \right),$$

where $\|\cdot\|$ is a standard Euclidean norm, and

$$A_{\text{cay}}(\omega) = \frac{2}{4 + \|\omega\|^2} (2I_3 + \hat{\omega}). \quad (53)$$

In the same way, after $u^{[p]}(t_{k+1})$ is obtained via evaluating $I(t_{k+1})$ termwise, we use (52) to update configuration $\bar{g}_{ab}(t_{k+1})$ at the endpoint t_{k+1} ,

$$\bar{g}_{ab}(t_{k+1}) = \bar{g}_{ab}(t_k) \cdot f(u^{[p]}(t_{k+1})). \quad (54)$$

3.2.2. Computing Velocities at the Gauss Points and the Endpoint. As mentioned above, here, we can directly use general pseudospectral in Section 3.1 to compute velocities at the Gauss points τ_1, \dots, τ_N and the endpoint t_{k+1} . However, due to (26) being implicit, we have to use iterative algorithm to solve it. When computing configurations in the preceding section, we assumed that $V_{ab}^b(\tau_i), i = 1, \dots, N$ in (46) are known in advance. Hence, we need to give their solutions, and we are going to do it in this section.

To begin with, we write (26) at all of the Gauss points in a compact matrix form,

$$\begin{aligned} & \begin{bmatrix} V_{ab}^{b(\text{new})}(\tau_1) \\ V_{ab}^{b(\text{new})}(\tau_2) \\ \vdots \\ V_{ab}^{b(\text{new})}(\tau_N) \end{bmatrix} \\ &= \frac{t_{k+1} - t_k}{2} \bar{D}^{-1} \\ & \cdot \begin{bmatrix} \Pi^{-1} \cdot \left(B(V_{ab}^{b(\text{old})}(\tau_1)) \cdot V_{ab}^{b(\text{old})}(\tau_1) + F(\tau_1, \bar{g}_{ab}^{(\text{old})}(\tau_1)) \right) \\ \Pi^{-1} \cdot \left(B(V_{ab}^{b(\text{old})}(\tau_2)) \cdot V_{ab}^{b(\text{old})}(\tau_2) + F(\tau_2, \bar{g}_{ab}^{(\text{old})}(\tau_2)) \right) \\ \vdots \\ \Pi^{-1} \cdot \left(B(V_{ab}^{b(\text{old})}(\tau_N)) \cdot V_{ab}^{b(\text{old})}(\tau_N) + F(\tau_N, \bar{g}_{ab}^{(\text{old})}(\tau_N)) \right) \end{bmatrix} \\ & - \bar{D}^{-1} \cdot \begin{bmatrix} d_{10} \\ d_{20} \\ \vdots \\ d_{N0} \end{bmatrix} \cdot V_{ab}^b(t_k), \end{aligned} \quad (55)$$

where $\bar{D} \in \mathbb{R}^{N \times N}$ is the submatrix of D in Section 3.1, and it is a reversible matrix,

$$\bar{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{bmatrix}. \quad (56)$$

By computing the velocity deviation between adjacent iterative steps, we determine whether or not to terminate the iterative process. It is noteworthy that we must ensure that computing configurations and velocities at the same Gauss points.

Finally, after the final satisfying iterative results of the velocities at the Gauss points are obtained, we use the following formula to compute velocity at the endpoint:

$$\begin{aligned} V_{ab}^b(t_{k+1}) &= V_{ab}^b(t_k) + \frac{t_{k+1} - t_k}{2} \\ & \times \sum_{i=1}^N \omega_i \cdot f\left(V_{ab}^{b(\text{new})}(\tau_i), \bar{g}_{ab}^{(\text{new})}(\tau_i)\right). \end{aligned} \quad (57)$$

3.2.3. 4th-Order Algorithm of Geometric Pseudospectral Method. In accordance with the aforementioned basic principle of geometric pseudospectral method, we develop a 4th-order algorithm for the rigid-body dynamics evolution over time.

Implementation and the Order Condition. Legendre-Gauss point τ_i is the root of Legendre polynomial $P_N(\tau) = (1/2^N N!)(d^N/d\tau^N)\{[\tau^2 - 1]^N\}$ of degree N . In fact, a convenient way to compute the Legendre-Gauss points is via the eigenvalues of the following tridiagonal Jacobi matrix:

$$J = \begin{bmatrix} 0 & \beta_1 & & & & \\ \beta_1 & 0 & \beta_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & 0 & \beta_{N-1} & \\ & & & \beta_{N-1} & 0 & \end{bmatrix}, \quad \beta_n = \frac{n}{\sqrt{4n^2 - 1}}. \quad (58)$$

Due to the fact that (25)-(26) are implicit, therefore, we have to use iterative algorithm to update the velocities and configurations at the Gauss points. In the interactive processes, we need to compute the velocity deviation and configuration deviation, respectively, between the adjacent iterative steps. For velocity deviation, since the Lie algebra space $\mathfrak{se}(3)$ in which the velocities belong is isomorphic to \mathbb{R}^6 , so we are able to use minus directly. However, for configuration deviation, because configuration space is a nonlinear manifold, we introduce the following natural error of configuration [27]:

$$\begin{aligned} e &= \left(\bar{g}_{ab}^{(\text{step}+1)}\right)^{-1} \bar{g}_{ab}^{(\text{step})} \\ &= \begin{bmatrix} \left(R^{(\text{step}+1)}\right)^T R^{(\text{step})} & \left(R^{(\text{step}+1)}\right)^T \left(p^{(\text{step})} - p^{(\text{step}+1)}\right) \\ 0 & 1 \end{bmatrix}. \end{aligned} \quad (59)$$

By the way, the reason why we choose the previous metric is its intuitive physical meaning; taking SE(2) as an example, the 2-norm $\|(R^{(\text{step}+1)})^T R^{(\text{step})}\|_2$ denotes angle difference $\Delta\theta = \theta^{(\text{step}+1)} - \theta^{(\text{step})}$ between configurations, and the 2-norm $\|(R^{(\text{step}+1)})^T (p^{(\text{step})} - p^{(\text{step}+1)})\|_2$ denotes distance difference $\Delta d = \sqrt{(x^{(\text{step}+1)} - x^{(\text{step})})^2 + (y^{(\text{step}+1)} - y^{(\text{step})})^2}$ between configurations. The last one required to pay attention to is to ensure computing configurations and velocities at the same Gauss points.

Turning to order condition of the algorithm, there are something worthy of our attention. Firstly, if we consider the time symmetry of the Magnus series expansion, the number of terms belong to (47) can be further reduced [13]. According to the time symmetry, function $u^{[p]}(t)$ can be expanded as the odd power of time t , and then $u^{[2q]}(t) = u_{2q-1}(t) + \mathcal{O}(t^{2q+1})$. For this purpose, we provide the top eight orders reduced truncated Magnus series expansion in Table 3 which have been used in step 2.2.3.3 and step 2.2.6.1 of Algorithm 1. Secondly, let the Magnus series expansion be truncated up to the p th-order trees for $\bar{g}_{ab}(t_{k+1})$ and $p-1$ th-order trees for the intermediate stages $\bar{g}_{ab}(\tau_i)$, then the resulting schemes have

p th-order [32]. Moreover, [2] stated that the interpolatory quadrature formula with N Gauss points is $2N$ th-order. In summary, our 4th-order geometric pseudospectral method only depends on two Gauss points.

3.3. Numerical Test on Free-Floating Rigid Body. In this section, the performance of Algorithm 1 is tested through evolution of the following free-floating rigid body dynamics equations (60)-(61) on $SE(3)$. Note that, under the condition of no external force, (60)-(61) are equivalent to (25)-(26). Table 1 summarizes the parameters used in the numerical test as follows:

$$\begin{bmatrix} \dot{R}_{ab}(t) & \dot{p}_{ab}^a(t) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{ab}(t) & p_{ab}^a(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\omega}_{ab}^b(t) & v_{ab}^b(t) \\ 0 & 0 \end{bmatrix}, \quad (60)$$

$$\begin{bmatrix} \dot{\omega}_{ab}^b(t) \\ \dot{v}_{ab}^b(t) \end{bmatrix} = \begin{bmatrix} J & 0 \\ 0 & mI_3 \end{bmatrix}^{-1} \begin{bmatrix} J\hat{\omega}_{ab}^b(t) & 0 \\ 0 & -m\hat{\omega}_{ab}^b(t) \end{bmatrix} \begin{bmatrix} \omega_{ab}^b(t) \\ v_{ab}^b(t) \end{bmatrix}. \quad (61)$$

For making the numerical test more comparable, we select four 4th-order numerical methods used to compare with geometric pseudospectral method (GPM), including ode45 [33], implicit 4th-order Runge-Kutta (RKi), explicit 4th-order RKMK (RKMK) [13], and 4th-order Gauss pseudospectral method (PM) [24], where iterative number and tolerance of the child loop of GPM and PM are set to 10 and 10^{-14} , respectively. Among the previous methods, ode45 is the build-in integrator from MATLAB which is a variable-step method and whose results would be used as the ground truth with a specified tolerance 10^{-14} . Both RKi and PM are the methods in Euclidean space. RKMK is an approximate Runge-Kutta method on Lie group [15]. It is worth mentioning that each test must be carried out under the same step size.

In order to intuitively present the difference between these methods, the 240 seconds evolving trajectories of the free-floating rigid body for a large step size are shown in Figure 1. As seen from the evolution versus time, the positions of PM is closest to the ground truth, and that of GPM take second place, but that of RKi and RKMK deviate from the true values to varying degrees. The potential reason resulted in accuracy of GPM being lower than PM is that position of PM is computed by iterative operator with specified iterative numbers and tolerance directly, but position of GPM is truncated to a specific order before iterative operator. Despite inferior to PM, the proposed method is obviously superior to RKi and RKMK in position accuracy. As can be seen, error accumulates faster in translational components p_{ab}^a than in rotational components R_{ab} . This could be explained by the fact that rotation is decoupled from translation and hence has its own error dynamics, whereas translation suffers from small cumulative errors in rotation. In addition, we also give the total computational time in the first line and the average time per step in parenthesis. Obviously, RKMK needs the least time since it is an explicit method. Both GPM and PM need less time than RKi. From orientation, linear velocity, and

angular velocity, there is no obvious difference, so we need to further compare these methods quantitatively.

In order to quantitatively compare the proposed method with the other three methods, deviation statistics and runtime statistics over 10 runs using a wide range of initial conditions on a 240 seconds evolution are provided in Figure 2. Figure 2(a) shows the position deviation against timesteps. It conforms that the accuracy of the GPM is superior to RKi and RKMK under the same large step size, although it is inferior to PM in average position deviation, and the relationship of position deviation between the four methods is consistent with that of Figure 1. In Figure 2(b), we give the orientation deviation against timesteps by calculating the metric $\|\log_{SO(3)}(R_{ode45}^T R_{ab})\|_2$ of $SO(3)$ [27], where R_{ode45} denotes the orientation of the free-floating rigid body which is computed by ode45. It should be noted that R_{ode45} transforms from the unit quaternion [26], since in ode45 unit quaternion is selected for parameterizing orientation at this point. Figure 2(c) shows the averaged computation time of four methods. As can be seen, computational efficiency of GPM is inferior to that of RKMK, because RKMK is an explicit method. However, our method requires less computation time than RKi and is nearly twice as efficient as RKi on average.

Finally, we examine the conservativeness of Lie group structure. Considering that configuration is parameterized by unit quaternion in ode45 and implicit RK method, we only compare the proposed method with explicit RKMK and Gauss pseudospectral method. As mentioned, the group element of $SO(3)$ satisfies $R_{ab}R_{ab}^T = I$, and we compute deviation $\|I_{3 \times 3} - R^T R\|_\infty$ for evaluating structural conservativeness of the algorithms [5]. Figure 3 illustrates that Gauss pseudospectral method in Euclidean space has no conservativeness of Lie group structure, since it is not a Lie group method; RKMK and geometric pseudospectral method on Lie group are able to preserve Lie group structure with accuracy approaching to machine accuracy.

4. Simulation and Control Application of Aircraft

It is seen from the numerical test in the previous section that the proposed method has better accuracy, stable structural conservativeness, and satisfying computational efficiency. Thus, it is able to meet the fidelity and timeliness requirements of aircraft dynamics simulation. In view of the aircraft dynamics is complex and underactuated, we adopt optimal control to generate flyable trajectories of aircraft. As mentioned in Section 1, pseudospectral method of the Gauss form can be easily applied to optimal control of the aircraft or other vehicles and satisfies optimization condition. Reference [23] shows that solving the NLP derived from the pseudospectral transcription of the Gauss form is exactly equivalent to solve discrete first-order necessary conditions. The KKT conditions of the NLP are exactly equivalent to the discrete first-order necessary conditions of the Bolza problem. And, KKT multipliers can be mapped to the costate of the continuous-time optimal control problem via costate

mapping principle. Because our geometric pseudospectral discretization scheme is also based on Gauss points, we can use it to transcribe a continuous aircraft optimal control problem into a discrete nonlinear programming problem (NLP). Then, the resulting NLP can be solved by many well-known optimization techniques [34, 35]. For this purpose, we follow the Gauss pseudospectral transcription approach introduced in [36]. The optimal control problem can be formulated as in Algorithm 2.

Depending on the specified scenario, cost functions include minimum control effort, minimum time, obstacle avoidance, and the quickest manoeuver. For further details regarding the optimization setup, one can consult to [36]. In this work, referring to the geometric optimal control framework proposed in [37], rotation component R_{ab} of g_{ab} is not treated as our decision variable, for three reasons: firstly, during the optimization, R_{ab} can be reconstructed from $\hat{\omega}_{ab}^b$ internally through evaluating (45), (51), and (54); secondly, the Lie group structural conservativeness could not be considered by NLP solvers, unless additional constraint with respect to structural conservativeness is introduced, which may affect optimality; moreover, computational efficiency is improved. We take obstacle avoidance as our example, with optimal control which is implemented using the combination of GPOPS [36] in MATLAB and SNOPT package in TOMLAB version 7.7 [38]. Figure 4 shows simulation results for an aircraft avoiding obstacles. Their detailed analysis is the subject of on-going work and will be presented in a future paper.

5. Conclusions

A completely left invariant rigid-body dynamics model of aircraft on $SE(3)$ is established. For the left invariance of rigid-body dynamics model in body-fixed frame, an equivalent differential equation on $\mathfrak{se}(3)$ is given under left trivialization. Accordingly, based on Magnus series expansion and exponential map, the solutions of configuration equation at the Gauss points and the endpoint are obtained. Velocities on $\mathfrak{se}(3)$ at the discrete points are computed based on general pseudospectral method, since $\mathfrak{se}(3)$ is isomorphic to \mathbb{R}^6 . Thereby, a 4th-order numerical method called geometric pseudospectral method is developed. Through a series of numerical tests and comparison with other numerical methods on Lie group and Euclidean space, the proposed method has a comprehensive advantage in accuracy, computational efficiency, and Lie group structural conservativeness. Finally, the way of applying our method to aircraft simulation and optimal control is illustrated.

For the future work, we will further analyze its performance in aircraft simulation and control. For presence of a large number of tedious commutator operator in multivariate quadrature, we will try to find an alternative method to simplify the calculation process so that extend our method to the higher order. Moreover, we will extend the proposed method to a broader kind of left invariant rigid-body dynamics systems in engineering.

Appendix

The top eight orders truncated Magnus series expansion of $u(t)$ under left trivialization (see Tables 2 and 3).

Acknowledgments

Great thanks are due to the reviewers for valuable comments on our revision of the paper. This work was supported by the National Natural Science Foundation of China (61005077), the National Basic Research Program of China (6138101001), and the National Defense Foundation of China (403060103).

References

- [1] N. Nordkvist and A. K. Sanyal, "A Lie group variational integrator for rigid body motion in $SE(3)$ with applications to underwater vehicle dynamics," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 5414–5419, Atlanta, Ga, USA, December 2010.
- [2] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, vol. 31 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2002.
- [3] T. Monovasilis, "Symplectic partitioned Runge-Kutta methods with the phase-lag property," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9075–9084, 2012.
- [4] T. Lee, M. Leok, and N. H. McClamroch, "Optimal attitude control of a rigid body using geometrically exact computations on $SO(3)$," *Journal of Dynamical and Control Systems*, vol. 14, no. 4, pp. 465–487, 2008.
- [5] T. Lee, M. Leok, and N. H. McClamroch, "A Lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum," in *Proceedings of the IEEE Conference on Control Applications*, pp. 962–967, 2005.
- [6] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, Fla, USA, 1994.
- [7] N. Bou-Rabee and J. E. Marsden, "Hamilton-Pontryagin integrators on Lie groups—part I: introduction and structure-preserving properties," *Foundations of Computational Mathematics*, vol. 9, no. 2, pp. 197–219, 2009.
- [8] T. E. Simos, "New open modified Newton Cotes type formulae as multilayer symplectic integrators," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 1983–1991, 2012.
- [9] T. E. Simos, "New stable closed Newton-Cotes trigonometrically fitted formulae for long-time integration," *Abstract and Applied Analysis*, vol. 2012, Article ID 182536, 15 pages, 2012.
- [10] T. Monovasilis, Z. Kalogiratou, and T. E. Simos, "Exponentially fitted symplectic Runge-Kutta-Nyström methods," *Applied Mathematics & Information Sciences*, vol. 7, no. 1, pp. 81–85, 2013.
- [11] Z. Kalogiratou, "Diagonally implicit trigonometrically fitted symplectic Runge-Kutta methods," *Applied Mathematics and Computation*, vol. 209, no. 14, pp. 7406–7412, 2013.
- [12] I. I. Hussein, M. Leok, A. K. Sanyal, and A. M. Bloch, "A discrete variational integrator for optimal control problems on $SO(3)$," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 6636–6641, December 2006.
- [13] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta Numerica*, vol. 9, pp. 215–365, 2000.

- [14] M. Kobilarov, K. Crane, and M. Desbrun, "Lie group integrators for animation and control of vehicles," *ACM Transactions on Graphics*, vol. 28, no. 2, pp. 1–14, 2009.
- [15] H. Munthe-Kaas, "Lie-Butcher theory for Runge-Kutta methods," *BIT*, vol. 35, no. 4, pp. 572–587, 1995.
- [16] P. E. Crouch and R. Grossman, "Numerical integration of ordinary differential equations on manifolds," *Journal of Nonlinear Science*, vol. 3, pp. 1–33, 1993.
- [17] J. C. Simo and L. Vu-Quoc, "On the dynamics in space of rods undergoing large motions—a geometrically exact approach," *Computer Methods in Applied Mechanics and Engineering*, vol. 66, no. 2, pp. 125–161, 1988.
- [18] E. Celledoni, A. Marthinsen, and B. Owren, "Commutator-free Lie group methods," Tech. Rep. 1/02, Norwegian University of Science and Technology, 2002.
- [19] T. Lee, M. Leok, and N. H. McClamroch, "Lie group variational integrators for the full body problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 29–30, pp. 2907–2924, 2007.
- [20] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, Pa, USA, 1995.
- [21] K. Engø, "On the construction of geometric integrators in the RKMK class," Tech. Rep. 158, Department of Computer Science, University of Bergen, 1998.
- [22] L. N. Trefethen, *Spectral Methods in Matlab*, SIAM, Philadelphia, Pa, USA, 2000.
- [23] D. A. Benson, *A Gauss pseudospectral transcription for optimal control [Ph.D. thesis]*, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2004.
- [24] A. V. Rao, "A survey of numerical methods for optimal control," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Pittsburgh, Pa, USA, August 2009, AAS paper 09-334.
- [25] R. Moulla, L. Lefèvre, and B. Maschke, "Geometric pseudospectral method for spatial integration of dynamical systems," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 17, no. 1, pp. 85–104, 2011.
- [26] B. Stevens and F. Lewis, *Aircraft Control and Simulation*, Wiley, New York, NY, USA, 1992.
- [27] F. Bullo and R. Murray, "Proportional derivative (PD) control on the Euclidean group," in *Proceedings of the European Control Conference*, Rome, Italy, 1995.
- [28] L. T. Nguyen, M. E. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown, and P. L. Deal, "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability," Tech. Rep. 1538, NASA, 1979.
- [29] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge University Press, 2007.
- [30] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [31] A. Iserles and S. P. Nørsett, "On the solution of linear differential equations in Lie groups," *Philosophical Transactions of the Royal Society A*, vol. 357, pp. 983–1020, 1999.
- [32] A. Zanna, "Collocation and relaxed collocation for the Fer and the Magnus expansions," *SIAM Journal on Numerical Analysis*, vol. 36, no. 4, pp. 1145–1182, 1999.
- [33] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [34] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: an SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, pp. 99–131, 2002.
- [35] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT Version 7: Software for Large Scale Non-Linear Programming*, 2006.
- [36] A. V. Rao, D. A. Benson, C. Darby et al., "Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, article 22, 2010.
- [37] M. Kobilarov, M. Desbrun, J. E. Marsden, and G. S. Sukhatme, "A discrete geometric optimal control framework for systems with symmetries," *Robotics: Science and Systems*, vol. 3, pp. 1–8, 2007.
- [38] K. Holmstrom, A. O. Goran, and M. M. Edvall, *User's Guide for TOMLAB/SNOPT*, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

