Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Mechanism and Machine Theory 48 (2012) 121–137

Contents lists available at SciVerse ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Lie group generalized-α time integration of constrained flexible multibody systems

Olivier Brüls [a,*], Alberto Cardona [b], Martin Arnold [c]

[a] University of Liège, Department of Aerospace and Mechanical Engineering (LTAS), Chemin des Chevreuils, 1 (B52), 4000 Liège, Belgium
[b] Universidad Nacional Litoral — Conicet, CIMEC-INTEC, Güemes 3450, 3000 Santa Fe, Argentina
[c] Martin Luther University Halle-Wittenberg, NWF II — Institute of Mathematics, 06099 Halle (Saale), Germany

ARTICLE INFO

Article history:
Received 27 January 2011
Received in revised form 25 July 2011
Accepted 28 July 2011
Available online 18 October 2011

Keywords:
Flexible multibody system
Time integration
Generalized-α method
Lie group
DAE
Index-3

ABSTRACT

This paper studies a Lie group extension of the generalized-α time integration method for the simulation of flexible multibody systems. The equations of motion are formulated as an index-3 differential-algebraic equation (DAE) on a Lie group, with the advantage that rotation variables can be taken into account without the need of introducing any parameterization. The proposed integrator is designed to solve this equation directly on the Lie group without index reduction. The convergence of the method for DAEs is studied in detail and global second-order accuracy is proven for all solution components, i.e. for nodal translations, rotations and Lagrange multipliers. The convergence properties are confirmed by three benchmarks of rigid and flexible systems with large rotation amplitudes. The Lie group method is compared with a more classical updated Lagrangian method which is also formulated in a Lie group setting. The remarkable simplicity of the new algorithm opens interesting perspectives for real-time applications, model-based control and optimization of multibody systems.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Today, the analysis of industrial flexible multibody systems can be efficiently and reliably achieved even by non-specialists using off the shelf simulation packages. More advanced studies may require the development of semi-analytical sensitivity analysis, structural optimization methods, real-time simulations and control design procedures which are generally not available in current simulation tools. Due to the underlying complexity of state of-the-art formulations in flexible multibody dynamics, the modification and extension of functionalities in existing simulation codes for such purposes is not a straightforward task. Therefore, the aim of the present work is to develop simpler but performant simulation methodologies for flexible multibody systems which are more suitable for the development of open platforms with extended functionalities.

The description of large rotations and the time integration problem are the main aspects reconsidered here for the development of simpler formulations in flexible multibody dynamics. According to the nonlinear finite element method, the configuration of an articulated system composed of rigid and flexible bodies is represented by a set of absolute nodal translation and rotation variables, see e.g. [1]. Each translation variable belongs to the linear space $\mathbb{R}^3$ whereas each rotation variable belongs to the nonlinear group of special orthogonal transformations $SO(3)$. In classical parameterization based approaches, rotation matrices, angular velocities as well as angular accelerations are explicitly represented in a coordinate system. The equations of motion are then written as differential-algebraic equations (DAE) in the vector space $\mathbb{R}^k$ so that they can be solved using standard DAE-solvers. In contrast, this paper shows that the formulation of the equations of motion in a vector space using an explicit

* Corresponding author. Tel.: +32 4 366 91 84; fax: +32 366 92 17.
E-mail addresses: o.bruls@ulg.ac.be (O. Brüls), acardona@intec.unl.edu.ar (A. Cardona), martin.arnold@mathematik.uni-halle.de (M. Arnold).

0094-114X/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechmachtheory.2011.07.017

Author's personal copy

parameterization of large rotations is not a necessary step for the simulation of flexible multibody systems and that significant simplifications are obtained if the equations of motion as well as the integration algorithm are directly formulated in a Lie group setting.

Let us briefly review classical parameterization-based simulation strategies. A first family of methods is based on a minimal parameterization, which means that three parameters are used to represent each rotation variable. Due to singularity problems, minimal parameterizations have a limited validity range and they can only be used locally. A reparameterization of the rotation manifold is needed when rotation amplitudes become too large. The updated Lagrangian point of view proposed in [1,2] is a kind of limit-case of this approach where the reparameterization is systematically planned at the end of each time step. In other words, the parameterization represents the incremental rotation with respect to the previous time step. A second family of methods is based on a global but redundant parameterization of rotations, which allows to avoid singularity problems at the cost of a larger set of equations to be solved. For example, the 4 Euler parameters can be used to represent an arbitrarily large rotation but one additional constraint should then be defined for each rotation variable [3]. Alternatively, the 9 components of the rotation matrix allow to represent a large rotation provided the definition of 6 additional constraints for each rotation variable [4]. The resulting equations of motion have the structure of a DAE on the linear parameter space, which can be solved using a suitable DAE solver.

Compared to classical parameterization-based methods, Lie group time integrators are designed to solve differential equations on Lie groups without any a priori definition of generalized coordinates. Crouch and Grossmann [5] and Munthe-Kaas [6,7] addressed some generalizations of classical Runge–Kutta and multistep time integration schemes for differential equations on Lie groups. By construction, those methods yield a numerical solution which inherently evolves on the manifold, without any drift-off phenomenon. In the Crouch and Grossman method, the exponential map is considered as the fundamental solution of differential equations on the Lie group, which allows to evaluate exactly the flow of "frozen" velocity fields. Hence, the numerical solution on the Lie group is computed by a suitable composition of elementary flows. In the Munthe-Kaas method, the exponential map is used to construct a local coordinate chart at each time step. This chart is defined about a particular point on the group which can be selected as the numerical solution at the previous time step. In this local coordinate system, a classical integration method can be used to compute the numerical solution. A close similarity is observed between the Munthe-Kaas approach and the updated Lagrangian method discussed above. Lie group methods have been applied for the analysis of rigid body dynamics [8] and nonlinear flexible beams [9].

The Lie group nature of rotational fields already played a major role in the development of geometrically consistent models for mechanical systems with large rotations [10–12]. This paper proposes a general DAE Lie group formulation of the equations of motion for flexible multibody systems. According to the nonlinear finite element method [1,13], the motion of a flexible multibody system is described on the Lie group defined by a multiple Cartesian product of $\mathbb{R}^3$ and $SO(3)$. The interconnections between the various bodies of the multibody system are modeled using nonlinear algebraic constraints. As a consequence, the motion of the system is restricted to a submanifold of the Lie group and the equations of motion have the structure of a DAE on a Lie group.

A family of Lie group time integrators was proposed in [14] to solve this DAE directly on the Lie group so that the equations of motion are not explicitly written in a local coordinate chart. The method is actually an extension of the generalized-α method for Lie group systems, which includes as special cases the classical generalized-α method for dynamic systems on a linear space [15] and the algorithm described in [10]. In the present paper the convergence of the method in the DAE case is studied in detail and second-order accuracy is proven for all solution components, i.e. for nodal translations, rotations and Lagrange multipliers. It is shown that the method inherits the favorable accuracy and stability properties of the classical generalized-α method and that it is a reliable approach for the simulation of complex multibody systems.

The proposed Lie group integrator is also compared with the updated Lagrangian method [1,2], which is reformulated here in a Lie group setting. The similarity between the updated Lagrangian approach and Munthe-Kaas Lie group method is highlighted. Numerical tests based on critical benchmarks for rigid and flexible multibody systems allow to compare the performance of the Lie group method and of the updated Lagrangian method from the viewpoint of accuracy and stability.

2. Equations of motion on a Lie group

The dynamics of a flexible multibody system can generally be described on a k-dimensional manifold G with a Lie group structure. From a mathematical viewpoint, a Lie group G is a differentiable manifold for which the product (or composition) and inversion operations are smooth maps, see [16]. In an absolute coordinate formulation, an element $q \in G$ is composed of several subsets of absolute nodal translations and rotations, a priori considered as independent variables. The composition operation $G \times G \to G$ is written as

$$q_{tot} = q_1 \circ q_2 \tag{1}$$

with $q_1, q_2, q_{tot} \in G$ and the identity element e is such that $q \circ e = e \circ q = q, \forall q \in G$. T_qG denotes the tangent space at a point $q \in G$ and the Lie algebra is defined as the tangent space at the identity $\mathfrak{g} = T_eG$. The Lie algebra is a vector space, which is isomorphic to $\mathbb{R}^k$ by an invertible linear mapping

$$\widetilde{(\bullet)}: \mathbb{R}^k \to \mathfrak{g}, \mathbf{v} \mapsto \tilde{\mathbf{v}}. \tag{2}$$

Author's personal copy

A tangent vector at any point q can be represented in the Lie algebra using the left translation map L_q. Indeed, L_q is a diffeomorphism of G

$$L_q\colon G\to G, y\mapsto q\circ y \tag{3}$$

and its derivative defines a diffeomorphism between T_yG and $T_{q\circ y}G$. In the particular case $y=e$, we thus have a bijection between $T_eG=\mathfrak{g}$ and T_qG:

$$DL_q(e)\colon \mathfrak{g}\to T_qG,\ \tilde{\mathbf{w}}\mapsto DL_q(e)\cdot\tilde{\mathbf{w}} \tag{4}$$

where $DL_q(e)\cdot\tilde{\mathbf{w}}$ is the directional derivative of L_q evaluated at point e in the direction $\tilde{\mathbf{w}}\in\mathfrak{g}$. Hence, a tangent vector $\tilde{\mathbf{w}}\in\mathfrak{g}$ defines a left invariant vector field on G which is constructed by left translation of $\tilde{\mathbf{w}}$ to the tangent space at any point of G.

In a multibody system, the nodal translation and rotation variables are generally not independent but they have to satisfy a set of m kinematic constraints $\Phi\colon G\to\mathbb{R}^m$, which restrict the dynamics to the submanifold N of dimension $k-m$

$$N=\{q\in G\colon \Phi(q)=\mathbf{0}\}. \tag{5}$$

Using classical principles of mechanics [1], the equations of motion of a flexible multibody system have the following index-3 differential-algebraic structure

$$\dot{q}=DL_q(e)\cdot\tilde{\mathbf{v}} \tag{6}$$

$$\mathbf{M}(q)\dot{\mathbf{v}}=-\mathbf{g}(q,\mathbf{v},t)-\mathbf{B}^T(q)\boldsymbol{\lambda} \tag{7}$$

$$\Phi(q)=\mathbf{0} \tag{8}$$

where $q\in G$ represents the configuration of the system, $\mathbf{v}\in\mathbb{R}^k$ is the velocity vector and $\boldsymbol{\lambda}\in\mathbb{R}^m$ is the vector of Lagrange multipliers associated with the constraints Φ. $\mathbf{M}$ is the $k\times k$ symmetric mass matrix, $\mathbf{g}$ is the vector of external, internal and complementary inertia forces and $\mathbf{B}$ is the $m\times k$ matrix of constraint gradients such that

$$D\Phi(q)\cdot\tilde{\mathbf{w}}=\mathbf{B}(q)\mathbf{w},\ \forall\mathbf{w}\in\mathbb{R}^k. \tag{9}$$

In the above equation, $D\Phi(q)\cdot\tilde{\mathbf{w}}$ is the directional derivative of Φ evaluated at point q in the direction of the tangent vector defined by left translation of $\tilde{\mathbf{w}}$, more precisely, it is equivalent to $D\Phi(q)\cdot\left(DL_q(e)\cdot\tilde{\mathbf{w}}\right)$. The equations of motion (6–8) allow to represent the dynamics of a general class of flexible multibody systems, e.g. using the finite element approach described in [1]. We observe that no parameterization of rotations is needed to formulate those equations.

In this formulation, velocities are represented using left invariant vector fields. Alternatively, velocities could also be represented using right invariant vector fields. In order to keep the mathematical developments compact, this alternative formulation is not developed in detail in the present paper.

2.1. Example 1: Single rotation system

For a dynamic system described by a single 3D rotation, the 3-dimensional Lie group G is simply $SO(3)$, the group of 3×3 proper orthogonal linear transformations. The composition operation is the matrix product $\mathbf{R}_1\circ\mathbf{R}_2=\mathbf{R}_1\mathbf{R}_2$ and the identity element is the 3×3 identity matrix $\mathbf{I}_3$. At any point $\mathbf{R}$, the tangent space is noted $T_{\mathbf{R}}SO(3)$ and the Lie algebra $\mathfrak{so}(3)=\left\{\tilde{\boldsymbol{\Omega}}\colon \tilde{\boldsymbol{\Omega}}+\tilde{\boldsymbol{\Omega}}^T=0\right\}$ is the set of skew-symmetric matrices. The Lie algebra can be identified to $\mathbb{R}^3$ since any matrix $\tilde{\boldsymbol{\Omega}}\in\mathfrak{so}(3)$

$$\tilde{\boldsymbol{\Omega}}=\begin{bmatrix}0 & -\Omega_3 & \Omega_2\\ \Omega_3 & 0 & -\Omega_1\\ -\Omega_2 & \Omega_1 & 0\end{bmatrix} \tag{10}$$

can be represented by the 3×1 axial vector $\boldsymbol{\Omega}=[\Omega_1\ \Omega_2\ \Omega_3]^T$. The tangent space $T_{\mathbf{R}}SO(3)$ is thus isomorphic to $\mathbb{R}^3$ and Eq. (6) becomes

$$\dot{\mathbf{R}}=DL_{\mathbf{R}}(\mathbf{I}_3)\cdot\tilde{\boldsymbol{\Omega}}=\mathbf{R}\tilde{\boldsymbol{\Omega}}. \tag{11}$$

From a physical point of view, $\boldsymbol{\Omega}$ is the vector of angular velocities in the body-attached (material) frame. Eq. (7) represents the equilibrium of angular momentum in the body-attached frame

$$\mathbf{J}\dot{\boldsymbol{\Omega}}+\boldsymbol{\Omega}\times\mathbf{J}\boldsymbol{\Omega}=\mathbf{C}(t) \tag{12}$$

Author's personal copy

with **J**, the inertia tensor in the body-attached frame, and $\mathbf{C}(t)$, the vector of applied external torque. There is no kinematic constraint in this example.

The alternative representation of velocities using right invariant vector fields would lead to the definition of $\boldsymbol{\omega}$, the vector of angular velocities in the inertial (spatial) frame. Both vectors are related by the frame transformation $\boldsymbol{\omega} = \mathbf{R}\boldsymbol{\Omega}$. Based on a right invariant representation, the equations of motion would be written in the inertial frame.

2.2. Example 2: System evolving on $\mathbb{R}^k$

The vector space $\mathbb{R}^k$ is a Lie group with the composition $\mathbf{q}_1 \circ \mathbf{q}_2 = \mathbf{q}_1 + \mathbf{q}_2$. Eq. (6) simply becomes $\dot{\mathbf{q}} = \mathbf{v}$.

2.3. Example 3: Rotation and translation of a single body system

The free motion of a single rigid body with combined translations and rotations can either be represented in $SE(3)$, the group of 4×4 homogeneous transformations, or in $\mathbb{R}^3 \times SO(3)$, the group formed by the Cartesian product of $\mathbb{R}^3$ and $SO(3)$. The second option is retained in the following, so that the configuration of the rigid body is represented by the pair $(\mathbf{x}, \mathbf{R})$ with the translation vector $\mathbf{x} \in \mathbb{R}^3$ and the rotation matrix $\mathbf{R} \in SO(3)$. Indeed, the set $\mathbb{R}^3 \times SO(3)$ is a 6-dimensional Lie group with the composition operation defined as $(\mathbf{x}_1, \mathbf{R}_1) \circ (\mathbf{x}_2, \mathbf{R}_2) = (\mathbf{x}_1 + \mathbf{x}_2, \mathbf{R}_1\mathbf{R}_2)$.

2.4. Example 4: Multibody system

The motion of a complex multibody system with k_n nodes is represented in the Lie group formed by the multiple Cartesian product of $\mathbb{R}^3$ and $SO(3)$

$$G = \mathbb{R}^3 \times \ldots \times \mathbb{R}^3 \times SO(3) \times \ldots \times SO(3). \tag{13}$$

Its elements have the form $(\mathbf{x}_1, \ldots, \mathbf{x}_{k_n}, \mathbf{R}_1 \ldots, \mathbf{R}_{k_n})$ with $\mathbf{x}_i \in \mathbb{R}^3$, the translation vector of node i, and $\mathbf{R}_i \in SO(3)$, the rotation matrix of node i. The composition operation is defined component-wise as in Example 3.

3. Exponential map

As seen in the previous section, the time derivative of the configuration variable q is conveniently represented in the Lie algebra using the left translation map of the group. The exponential map, which maps any element of the Lie algebra to the Lie group

$$\exp: \mathfrak{g} \to G, \quad \tilde{\mathbf{q}} \mapsto q = \exp(\tilde{\mathbf{q}}) \tag{14}$$

is a useful tool for the design of Lie group integrators. Its mathematical definition is related with integral curves of left (or right) invariant vector fields. Accordingly, the solution of the fundamental equation

$$\dot{q}(t) = DL_q(e) \cdot \tilde{\mathbf{w}}, \quad q(0) = q_0 \tag{15}$$

for a constant $\tilde{\mathbf{w}} \in \mathfrak{g}$ is given by

$$q(t) = q_0 \circ \exp(t\tilde{\mathbf{w}}). \tag{16}$$

The exponential map admits the series expansion

$$\exp(\tilde{\mathbf{q}}) = \sum_{i=0}^{\infty} \frac{1}{i!} \tilde{\mathbf{q}}^i. \tag{17}$$

However, this equation only holds for matrix Lie groups, i.e. Lie groups whose elements can be represented as matrices, see [17] for more details. It can easily be seen that all Lie groups considered in this paper for the analysis of multibody systems are actually matrix Lie groups. The exponential map allows the construction of a local parameterization of G about an arbitrary point $q_0 \in G$. Indeed, the relation

$$q = q_0 \circ \exp(\tilde{\mathbf{q}}) \tag{18}$$

defines a diffeomorphism between G and $\mathfrak{g}$ and consequently between G and $\mathbb{R}^k$. This diffeomorphism can be written as a coordinate map $\mathbb{R}^k \to G$: $\mathbf{q} \mapsto q = q_0 \circ \exp(\tilde{\mathbf{q}})$.

Author's personal copy

The derivative of the exponential map can be obtained by differentiation of Eq. (18) with respect to time

$$\dot{q} = q_o \circ \left(D\exp(\tilde{\mathbf{q}}) \cdot \dot{\tilde{\mathbf{q}}} \right). \tag{19}$$

This equation defines a relation between $\dot{q}$ and the time derivative of the Lie algebra coordinates $\dot{\mathbf{q}}$. Using Eqs. (6) and (18), $\dot{q}$ can also be written in terms of the left invariant derivative

$$\dot{q} = q_o \circ \left(DL_{\exp(\tilde{\mathbf{q}})}(e) \cdot \tilde{\mathbf{v}} \right) \tag{20}$$

A comparison of the last two equations leads to a linear relationship between $\dot{\mathbf{q}}$ and $\mathbf{v}$

$$D\exp(\tilde{\mathbf{q}}) \cdot \dot{\tilde{\mathbf{q}}} = DL_{\exp(\tilde{\mathbf{q}})}(e) \cdot \tilde{\mathbf{v}}. \tag{21}$$

In literature, this relation is sometimes written as

$$d\exp_{\tilde{\mathbf{q}}} : \mathfrak{g} \to \mathfrak{g}, \quad \dot{\tilde{\mathbf{q}}} \mapsto \tilde{\mathbf{v}} \tag{22}$$

or as a linear relation from $\mathbb{R}^k$ to $\mathbb{R}^k$

$$\mathbf{v} = \mathbf{T}(\mathbf{q})\dot{\mathbf{q}} \tag{23}$$

where $\mathbf{T}(\mathbf{q})$ is the so-called tangent operator of the exponential map. For a matrix group, it admits the series expansion [17]

$$\mathbf{T}(\mathbf{q}) = \sum_{i=0}^{\infty} \frac{(-1)^i}{(i+1)!} \tilde{\mathbf{q}}^i. \tag{24}$$

For $G = SO(3)$, the exponential operator can be computed using the Rodrigues formula:

$$\exp_{SO(3)}\left(\tilde{\psi}\right) = \mathbf{I}_3 + \frac{\sin\phi}{\phi}\tilde{\psi} + \frac{1-\cos\phi}{\phi^2}\tilde{\psi}\tilde{\psi} \tag{25}$$

where the axial vector $\psi \in \mathbb{R}^3$ is the so-called Cartesian rotation vector and $\phi = \|\psi\|$. The tangent operator $\mathbf{T}_{SO(3)}(\psi)$, which appears in the relation $\boldsymbol{\Omega} = \mathbf{T}_{SO(3)}(\psi)\dot{\psi}$ between the angular velocity $\boldsymbol{\Omega}$ and the vector $\dot{\psi}$, is given by the formula

$$\mathbf{T}_{SO(3)}(\psi) = \mathbf{I}_3 + \frac{cos\phi - 1}{\phi^2}\tilde{\psi} + \left(1 - \frac{sin\phi}{\phi}\right)\frac{\tilde{\psi}\tilde{\psi}}{\phi^2}. \tag{26}$$

For $G = \mathbb{R}^k$, the exponential map is the identity $\exp_{\mathbb{R}^k}(\mathbf{q}) = \mathbf{q}$ and the tangent operator is the identity $\mathbf{T}_{\mathbb{R}^k}(\mathbf{q}) = \mathbf{I}_k$.

For $G = \mathbb{R}^3 \times SO(3)$, the exponential map (as well as the tangent operator) is computed independently for the translation and rotation variables: $\exp_{\mathbb{R}^3 \times SO(3)}\left(\mathbf{x}, \tilde{\psi}\right) = \left(\exp_{\mathbb{R}^3}(\mathbf{x}), \exp_{SO(3)}\left(\tilde{\psi}\right)\right)$. This definition is easily extended to general multibody systems with an arbitrary number of translation and rotation variables. In such a way, the exponential map and the tangent operator can be evaluated using analytical formulae in the general case.

4. Lie group time integrator

Inspired by the index-3 formulation of the generalized-α time integration scheme for classical systems of differential-algebraic Eq. [18] as well as by the work of Crouch and Grossmann [5] and Munthe-Kaas [6,7], we propose a family of Lie group time integrators based on the following discretized set of equations:

$$\mathbf{M}(q_{n+1})\dot{\mathbf{v}}_{n+1} = -\mathbf{g}(q_{n+1}, \mathbf{v}_{n+1}, t_{n+1}) - \mathbf{B}^T(q_{n+1})\boldsymbol{\lambda}_{n+1} \tag{27}$$

$$\boldsymbol{\Phi}(q_{n+1}) = \mathbf{0} \tag{28}$$

$$q_{n+1} = q_n \circ \exp\left(h\widetilde{\Delta\mathbf{q}_n}\right) \tag{29}$$

$$\Delta\mathbf{q}_n = \mathbf{v}_n + (0.5-\beta)h\mathbf{a}_n + \beta h\mathbf{a}_{n+1} \tag{30}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + (1-\gamma)h\mathbf{a}_n + \gamma h\mathbf{a}_{n+1} \tag{31}$$

Author's personal copy

$$(1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m \mathbf{a}_n = \left(1-\alpha_f\right)\dot{\mathbf{v}}_{n+1} + \alpha_f \dot{\mathbf{v}}_n \tag{32}$$

where $h = t_{n+1} - t_n$ is the time step size. The variable $\widetilde{\Delta \mathbf{q}_n}$ may be interpreted as an average and "frozen" velocity field in the time interval between t_n and t_{n+1}. The particular form of Eq. (29) makes this formulation applicable to dynamic systems on Lie groups. Let us note that additional similar Lie group schemes are discussed in [14], which are based on alternative formulations of Eqs. (29) and (30).

The proposed integration scheme has several interesting properties.

1. The equations of motion are enforced exactly at time $n+1$ and there is no weighted combination of forces between time n and time $n+1$. As a consequence, the Lagrange multipliers $\boldsymbol{\lambda}_n$ are not involved in the computation of step $n+1$.
2. The variable $\mathbf{a}_{n+1}$ is an acceleration-like variable which is different from the true acceleration $\dot{\mathbf{v}}_{n+1}$ at time t_{n+1}. Using a first order approximation, it can be initialized as $\mathbf{a}_0 = \dot{\mathbf{v}}_0$.
3. The derivative $\dot{q} \in T_q G$ is never explicitly evaluated in the numerical procedure. The algorithm only involves operations on tangent vectors in the Lie algebra $\mathfrak{g}$.
4. Nonlinearities are present not only in the equations of motion, but also in the integration formula (29).
5. The algorithm includes as a special case the classical generalized-α algorithm described in [18].

In the step-by-step integration procedure, Eqs. (27–32) are solved for all variables at time step $n+1$ based on their value at time step n using the Newton–Raphson algorithm in Table 1. In this algorithm, the $k \times 1$ residual vector is defined as

$$\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) = \mathbf{M}(q)\dot{\mathbf{v}} + \mathbf{g}(q, \mathbf{v}, t) + \mathbf{B}^T(q)\boldsymbol{\lambda} \tag{33}$$

and the variables $\beta' = (1-\alpha_m)/(\beta h^2(1-\alpha_f))$ and $\gamma' = \gamma/\beta h$ are algorithmic parameters. The $(k+m)\times(k+m)$ iteration matrix $\mathbf{S}_t(q, h\Delta\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, t)$ is evaluated as

$$\mathbf{S}_t = \begin{bmatrix} \mathbf{M}(q)\beta' + \mathbf{C}_t(q, \mathbf{v}, t)\gamma' + \mathbf{K}_t(q, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, t)\mathbf{T}(h\Delta\mathbf{q}) & \mathbf{B}^T(q) \\ \mathbf{B}(q)\mathbf{T}(h\Delta\mathbf{q}) & \mathbf{0} \end{bmatrix}. \tag{34}$$

In this expression, $\mathbf{K}_t$ and $\mathbf{C}_t$ are the tangent stiffness and damping matrices such that $D_1\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) \cdot \tilde{\mathbf{w}} = \mathbf{K}_t\mathbf{w}$ and $D_2\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) \cdot \mathbf{w} = \mathbf{C}_t\mathbf{w}\ \forall\, \mathbf{w} \in \mathbb{R}^k$, where D_1 (respectively, D_2) indicates the directional derivative with respect to the first argument $q \in G$ (respectively, to the second argument $\mathbf{v} \in \mathbb{R}^k$). For small time steps h, the matrix $\mathbf{S}_t$ becomes severely ill conditioned. This difficulty can be eliminated by the implementation of a suitable scaling strategy, see e.g. [18–20]. Compared to classical parameterization-based methods, one also observes that the iteration matrix is much simpler here so that the implementation of its exact expression in a simulation code is more easily feasible.

Table 1
Numerical algorithm for a single time step.

function solveTimeStep$(q_n, \mathbf{v}_n, \dot{\mathbf{v}}_n, \mathbf{a}_n)$
  $\dot{\mathbf{v}}_{n+1} := \mathbf{0}$
  $\boldsymbol{\lambda}_{n+1} := \mathbf{0}$
  $\mathbf{a}_{n+1} := (\alpha_f \dot{\mathbf{v}}_n - \alpha_m \mathbf{a}_n)/(1-\alpha_m)$
  $\mathbf{v}_{n+1} := \mathbf{v}_n + h(1-\gamma)\mathbf{a}_n + \gamma h \mathbf{a}_{n+1}$
  $\Delta\mathbf{q}_n := \mathbf{v}_n + (0.5-\beta)h\mathbf{a}_n + \beta h \mathbf{a}_{n+1}$
  for $i = 1$ to i_{max}
    $q_{n+1} := q_n \circ \exp(h\widetilde{\Delta\mathbf{q}_n})$
    $\mathbf{res} := \begin{bmatrix} \mathbf{r}(q_{n+1}, \mathbf{v}_{n+1}, \boldsymbol{\lambda}_{n+1}, \dot{\mathbf{v}}_{n+1}, t_{n+1}) \\ \boldsymbol{\Phi}(q_{n+1}) \end{bmatrix}$
    if $\|\mathbf{r}\| < \mathrm{tol}_{\mathbf{r}}$ and $\|\boldsymbol{\Phi}\| < \mathrm{tol}_{\boldsymbol{\Phi}}$
      break
    end
    $\mathbf{S}_t := \mathbf{S}_t(q_{n+1}, h\Delta\mathbf{q}_n, \mathbf{v}_{n+1}, \dot{\mathbf{v}}_{n+1}, \boldsymbol{\lambda}_{n+1}, t_{n+1})$
    $\begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} := -\mathbf{S}_t^{-1}\mathbf{res}$
    $\Delta\mathbf{q}_n := \Delta\mathbf{q}_n + \Delta\mathbf{x}/h$
    $\mathbf{v}_{n+1} := \mathbf{v}_{n+1} + \gamma'\Delta\mathbf{x}$
    $\dot{\mathbf{v}}_{n+1} := \dot{\mathbf{v}}_{n+1} + \beta'\Delta\mathbf{x}$
    $\boldsymbol{\lambda}_{n+1} := \boldsymbol{\lambda}_{n+1} + \Delta\boldsymbol{\lambda}$
  end
  $\mathbf{a}_{n+1} := \mathbf{a}_{n+1} + (1-\alpha_f)/(1-\alpha_m)\dot{\mathbf{v}}_{n+1}$
return $q_{n+1}, \mathbf{v}_{n+1}, \dot{\mathbf{v}}_{n+1}, \boldsymbol{\lambda}_{n+1}, \mathbf{a}_{n+1}$

Author's personal copy

5. Convergence of the proposed Lie group time integrator

Inspired by the convergence analysis of multi step methods for higher index DAEs [21], second order convergence of the generalized-α method for constrained systems was shown in [18]. In this section, this result is generalized to the Lie group integrator defining local and global errors as elements of the corresponding Lie algebra [22].

The inherent nonlinearity of the Lie group method introduces additional higher order error terms that are analyzed in detail using the Baker–Campbell–Hausdorff formula for the product of matrix exponentials ([23], Section III.4.2). For commuting matrices $\mathbf{A}$, $\mathbf{B}$, we have $\exp(\mathbf{A})\circ\exp(\mathbf{B})=\exp(\mathbf{A}+\mathbf{B})$. In the general case, additional terms containing commutators $[\mathbf{A},\mathbf{B}]:=\mathbf{AB}-\mathbf{BA}$ have to be considered. For matrices $\mathbf{A}=\mathcal{O}(h)$, $\mathbf{B}=\mathcal{O}(h)$, we get

$$\exp(\mathbf{A})\circ\exp(\mathbf{B}) = \exp\left(\mathbf{A}+\mathbf{B}+\frac{1}{2}[\mathbf{A},\mathbf{B}]+\mathcal{O}(h)\|[\mathbf{A},\mathbf{B}]\|\right), \tag{35}$$

see ([23], Eq. (III.4.11)). Assuming matrices $\mathbf{A}=\mathcal{O}(h^{r_\mathbf{A}})$, $\mathbf{B}=\mathcal{O}(h^{r_\mathbf{B}})$ with $\mathbf{A}-\mathbf{B}=\mathcal{O}(h^{r_{\mathbf{A}-\mathbf{B}}})$ and using the identity $[\mathbf{A},\mathbf{B}]=-[\mathbf{A},\mathbf{A}-\mathbf{B}]=[\mathbf{A}-\mathbf{B},\mathbf{B}]$, estimate (35) results in

$$\exp(\mathbf{A})\circ\exp(\mathbf{B}) = \exp(\mathbf{A}+\mathbf{B}+\mathcal{O}(h^r)) \tag{36}$$

with $r:=\max\{r_\mathbf{A}+r_\mathbf{B}, r_\mathbf{A}+r_{\mathbf{A}-\mathbf{B}}, r_\mathbf{B}+r_{\mathbf{A}-\mathbf{B}}\}$.

5.1. Local error

The local errors $\mathbf{l}_n^{(\cdot)}$ in solution components $q\in G$, $\mathbf{v}\in\mathbb{R}^k$ and $\mathbf{a}\in\mathbb{R}^k$ are defined similar to the local error of one step methods for ODEs [21] and for differential equations on Lie groups [22]:

$$\hat{q}_{n+1} = q(t_{n+1})\circ\exp\left(\tilde{\mathbf{l}}_n^q\right),\quad \hat{\mathbf{v}}_{n+1} = \mathbf{v}(t_{n+1})+\mathbf{l}_n^v,\quad \hat{\mathbf{a}}_{n+1} = \dot{\mathbf{v}}(t_{n+1}+\Delta_\alpha h)+\mathbf{l}_n^a \tag{37}$$

with $\hat{q}_{n+1}:=q(t_n)\circ\exp\left(h\widetilde{\Delta\mathbf{q}}(t_n)\right)$, $\Delta_\alpha h:=(\alpha_m-\alpha_f)h$ and

$$\Delta\mathbf{q}(t_n) := \mathbf{v}(t_n)+h(0.5-\beta)\dot{\mathbf{v}}(t_n+\Delta_\alpha h)+h\beta\dot{\mathbf{v}}(t_{n+1}+\Delta_\alpha h) \tag{38}$$

$$\hat{\mathbf{v}}_{n+1} := \mathbf{v}(t_n)+h(1-\gamma)\dot{\mathbf{v}}(t_n+\Delta_\alpha h)+h\gamma\dot{\mathbf{v}}(t_{n+1}+\Delta_\alpha h) \tag{39}$$

$$(1-\alpha_m)\hat{\mathbf{a}}_{n+1}+\alpha_m\dot{\mathbf{v}}(t_n+\Delta_\alpha h) = \left(1-\alpha_f\right)\dot{\mathbf{v}}(t_{n+1})+\alpha_f\dot{\mathbf{v}}(t_n). \tag{40}$$

At the right hand sides of Eqs. (37), (38) and (39) and at the left hand side of Eq. (40), the arguments of $\dot{\mathbf{v}}$ are shifted by $\Delta_\alpha h$ to guarantee the local error estimate $\mathbf{l}_n^a=\mathcal{O}\left(h^2\right)$ which is a straightforward consequence of

$$(1-\alpha_m)\mathbf{l}_n^a = \left(\left(1-\alpha_f\right)\dot{\mathbf{v}}(t_{n+1})+\alpha_f\dot{\mathbf{v}}(t_n)\right)-\left((1-\alpha_m)\dot{\mathbf{v}}(t_{n+1}+\Delta_\alpha h)+\alpha_m\dot{\mathbf{v}}(t_n+\Delta_\alpha h)\right). \tag{41}$$

In other words, $\hat{\mathbf{a}}_n$ is a second order approximation to $\dot{\mathbf{v}}(t_n+\Delta_\alpha h)$. With the standard second-order condition $\gamma=0.5+\alpha_f-\alpha_m$ for generalized-α methods [15], Taylor expansion of $\mathbf{l}_n^v$ results in

$$\mathbf{l}_n^v = h^2\left(\gamma-\alpha_f+\alpha_m-0.5\right)\ddot{\mathbf{v}}(t_n)+\mathcal{O}\left(h^3\right)=\mathcal{O}\left(h^3\right). \tag{42}$$

The local error analysis for solution components $q\in G$ is based on the Taylor expansion of $h\widetilde{\Delta\mathbf{q}}(t_n)\in\mathfrak{g}$ in the definition of $\hat{q}_{n+1}$, see Eq. (38), and a corresponding expression for the analytical solution $q(t_{n+1})$. If $\tilde{\mathbf{v}}(t)$ in Eq. (6) is given, then we get

$$q(t_{n+1}) = q(t_n+h) = q(t_n)\circ\exp\left(h\tilde{\mathbf{v}}(h;t_n,q(t_n))\right) \tag{43}$$

with a smooth function $\tilde{\mathbf{v}}$: $[-h_0,h_0]\times\mathbb{R}\times G\to\mathfrak{g}$. With this notation, an explicit expression for $\exp\left(\tilde{\mathbf{l}}_n^q\right)$ in Eq. (37) is obtained:

$$\exp\left(\tilde{\mathbf{l}}_n^q\right) = \left(q(t_{n+1})\right)^{-1}\circ q(t_n)\circ\exp\left(h\widetilde{\Delta\mathbf{q}}(t_n)\right) = \exp\left(-h\tilde{\mathbf{v}}(h;t_n,q(t_n))\right)\circ\exp\left(h\widetilde{\Delta\mathbf{q}}(t_n)\right). \tag{44}$$

Following the approach of Müller [24], the Magnus expansion of $h\tilde{\mathbf{v}}$, see [23], may be used to show

$$h\tilde{\mathbf{v}}\left(h;t_n,q(t_n)\right) = h\tilde{\mathbf{v}}(t_n)+\frac{h^2}{2}\dot{\tilde{\mathbf{v}}}(t_n)+\frac{h^3}{6}\ddot{\tilde{\mathbf{v}}}(t_n)+\frac{h^3}{12}\left[\tilde{\mathbf{v}}(t_n),\dot{\tilde{\mathbf{v}}}(t_n)\right]+\mathcal{O}\left(h^4\right) \tag{45}$$

$$= h\widetilde{\Delta\mathbf{q}}(t_n)+\frac{h^3}{6}\left(\left(1-6\beta-3\left(\alpha_m-\alpha_f\right)\right)\ddot{\tilde{\mathbf{v}}}(t_n)+\frac{1}{2}\left[\tilde{\mathbf{v}}(t_n),\dot{\tilde{\mathbf{v}}}(t_n)\right]\right)+\mathcal{O}\left(h^4\right) \tag{46}$$

Author's personal copy

which coincides for the translational degrees of freedom (as in the classical ODE case) with a standard Taylor expansion of the analytical solution since the Lie bracket [.,.] vanishes in this special case. Applying the Baker–Campbell–Hausdorff formula, see Eqs. (36) to (44) we have $r_{\mathbf{A}}=r_{\mathbf{B}}=1$, $r_{\mathbf{A}-\mathbf{B}}=3$ and

$$\tilde{\mathbf{l}}_n^q = h\left(\widetilde{\Delta\mathbf{q}}(t_n) - \tilde{\mathbf{v}}\left(h; t_n, q(t_n)\right)\right) + \mathcal{O}\left(h^4\right) = \mathcal{O}\left(h^3\right)$$

and

$$\tilde{\mathbf{l}}_{n-1}^q = \frac{h^3}{6}\left(\left(6\beta + 3\left(\alpha_m-\alpha_f\right)-1\right)\dot{\tilde{\mathbf{v}}}(t_{n-1}) - \frac{1}{2}\left[\tilde{\mathbf{v}}(t_{n-1}), \dot{\tilde{\mathbf{v}}}(t_{n-1})\right]\right) + \mathcal{O}\left(h^4\right) = \tilde{\mathbf{l}}_n^q + \mathcal{O}\left(h^4\right). \tag{47}$$

5.2. Global error

In the Lie group integrator for constrained systems, the error propagation in the differential components q, $\mathbf{v}$ is coupled with the error propagation in the algebraic components $\mathbf{a}$, $\boldsymbol{\lambda}$. Similar to Eq. (37), the global errors $\mathbf{e}_n^{(\bullet)}$ are defined by

$$q_n = q(t_n)\circ\exp(\tilde{\mathbf{e}}_n^q), \quad \mathbf{v}_n = \mathbf{v}(t_n) + \mathbf{e}_n^{\mathbf{v}}, \quad \mathbf{a}_n = \dot{\mathbf{v}}\left(t_n + \left(\alpha_m-\alpha_f\right)h\right) + \mathbf{e}_n^{a}, \quad \boldsymbol{\lambda}_n = \boldsymbol{\lambda}(t_n) + \mathbf{e}_n^{\boldsymbol{\lambda}}. \tag{48}$$

Furthermore, the notation $\mathbf{e}_n^{\dot{\mathbf{v}}}$ with $\dot{\mathbf{v}}_n = \dot{\mathbf{v}}(t_n) + \mathbf{e}_n^{\dot{\mathbf{v}}}$ will be useful. Because of Eqs. (7) and (27) at $t=t_n$, this term is bounded by

$$\left\|\mathbf{e}_n^{\dot{\mathbf{v}}}\right\| = \mathcal{O}(1)\left(\left\|\mathbf{e}_n^q\right\| + \left\|\mathbf{e}_n^{\mathbf{v}}\right\| + \left\|\mathbf{e}_n^{\boldsymbol{\lambda}}\right\|\right). \tag{49}$$

Multiplying Eq. (7) at $t=t_n$ and Eq. (27) from the left by $[(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)^{-1}\mathbf{B}\mathbf{M}^{-1}](q(t_n))$, we get

$$\mathbf{e}_n^{\boldsymbol{\lambda}} = -\left[\left(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\right)^{-1}\right]\left(q(t_n)\right)\cdot\mathbf{B}(q(t_n))\mathbf{e}_n^{\dot{\mathbf{v}}} + \mathcal{O}(h)\left\|\mathbf{e}_n^{\dot{\mathbf{v}}}\right\| + \mathcal{O}(1)\left(\left\|\mathbf{e}_n^q\right\| + \left\|\mathbf{e}_n^{\mathbf{v}}\right\|\right) + \mathcal{O}(h)\left\|\mathbf{e}_n^{\boldsymbol{\lambda}}\right\|. \tag{50}$$

Here and in the following we suppose $\left\|\mathbf{e}_n^q\right\| = \mathcal{O}(h)$, $\left\|\mathbf{e}_n^{\mathbf{v}}\right\| = \mathcal{O}(h)$ and $\left\|\mathbf{e}_n^{\mathbf{a}}\right\| = \mathcal{O}(1)$. For a second order method, this technical assumption may be verified by standard arguments [21].

5.3. Error propagation in the Lie group and its tangent space

The error propagation in the solution components $\mathbf{v}$ may be studied similar to one step methods for ODEs:

$$\mathbf{e}_{n+1}^{\mathbf{v}} = \mathbf{v}_{n+1} - \mathbf{v}(t_{n+1}) = (\mathbf{v}_{n+1} - \hat{\mathbf{v}}_{n+1}) + (\hat{\mathbf{v}}_{n+1} - \mathbf{v}(t_{n+1})) = \mathbf{e}_n^{\mathbf{v}} + h(1-\gamma)\mathbf{e}_n^{\mathbf{a}} + h\gamma\mathbf{e}_{n+1}^{\mathbf{a}} + \mathbf{l}_n^v. \tag{51}$$

For component $\mathbf{a}$, we get

$$\begin{aligned}(1-\alpha_m)\mathbf{e}_{n+1}^{\mathbf{a}} + \alpha_m\mathbf{e}_n^{\mathbf{a}} &= (1-\alpha_m)(\mathbf{a}_{n+1} - \hat{\mathbf{a}}_{n+1}) + (1-\alpha_m)\left(\hat{\mathbf{a}}_{n+1} - \dot{\mathbf{v}}\left(t_{n+1} + \left(\alpha_m-\alpha_f\right)h\right)\right) + \alpha_m\left(\mathbf{a}_n - \dot{\mathbf{v}}\left(t_n + \left(\alpha_m-\alpha_f\right)h\right)\right)\\ &= (1-\alpha_m)\mathbf{l}_n^{\mathbf{a}} + \left(1-\alpha_f\right)\mathbf{e}_{n+1}^{\dot{\mathbf{v}}} + \alpha_f\mathbf{e}_n^{\dot{\mathbf{v}}} = \mathcal{O}\left(h^2\right) + \mathcal{O}(1)\sum_{k=0}^{1}\left(\left\|\mathbf{e}_{n+k}^q\right\| + \left\|\mathbf{e}_{n+k}^{\mathbf{v}}\right\| + \left\|\mathbf{e}_{n+k}^{\boldsymbol{\lambda}}\right\|\right),\end{aligned} \tag{52}$$

see Eq. (49). This estimate can be used to eliminate $\mathbf{e}_n^{\mathbf{a}}$, $\mathbf{e}_{n+1}^{\mathbf{a}}$ in Eq. (51) considering a weighted linear combination of two subsequent time steps [18], see also [25]:

$$\begin{aligned}(1-\alpha_m)\mathbf{e}_{n+1}^{\mathbf{v}} - (1-2\alpha_m)\mathbf{e}_n^{\mathbf{v}} - \alpha_m\mathbf{e}_{n-1}^{\mathbf{v}} &= (1-\alpha_m)(\mathbf{e}_{n+1}^{\mathbf{v}} - \mathbf{e}_n^{\mathbf{v}}) + \alpha_m(\mathbf{e}_n^{\mathbf{v}} - \mathbf{e}_{n-1}^{\mathbf{v}})\\ &= h(1-\gamma)\left((1-\alpha_m)\mathbf{e}_n^{\mathbf{a}} + \alpha_m\mathbf{e}_{n-1}^{\mathbf{a}}\right) + h\gamma\left((1-\alpha_m)\mathbf{e}_{n+1}^{\mathbf{a}} + \alpha_m\mathbf{e}_n^{\mathbf{a}}\right) + \mathcal{O}\left(h^3\right)\\ &= h(1-\gamma)\left(\left(1-\alpha_f\right)\mathbf{e}_n^{\dot{\mathbf{v}}} + \alpha_f\mathbf{e}_{n-1}^{\dot{\mathbf{v}}}\right) + h\gamma\left(\left(1-\alpha_f\right)\mathbf{e}_{n+1}^{\dot{\mathbf{v}}} + \alpha_f\mathbf{e}_n^{\dot{\mathbf{v}}}\right) + \mathcal{O}\left(h^3\right)\end{aligned} \tag{53}$$

$$= \mathcal{O}(h)\sum_{k=0}^{2}\left(\left\|\mathbf{e}_{n+k-1}^q\right\| + \left\|\mathbf{e}_{n+k-1}^{\mathbf{v}}\right\| + \left\|\mathbf{e}_{n+k-1}^{\boldsymbol{\lambda}}\right\|\right) + \mathcal{O}\left(h^3\right). \tag{54}$$

To get an expression like Eq. (51) for the solution components $q(t)$, we observe that $q(t_n)\circ\exp\left(h\,\widetilde{\Delta\mathbf{q}}(t_n)\right) = \hat{q}_{n+1} = q(t_{n+1})\circ\exp\left(\tilde{\mathbf{l}}_n^q\right)$, i.e.,

$$\begin{aligned}\exp\left(\tilde{\mathbf{l}}_n^q\right)\circ\exp\left(-h\,\widetilde{\Delta\mathbf{q}}(t_n)\right) &= \left(q(t_{n+1})\right)^{-1}\circ q(t_n) = \left(q_{n+1}\circ\exp(-\tilde{\mathbf{e}}_{n+1}^q)\right)^{-1}\circ\left(q_n\circ\exp(-\tilde{\mathbf{e}}_n^q)\right)\\ &= \exp(\tilde{\mathbf{e}}_{n+1}^q)\circ q_{n+1}^{-1}\circ q_n\circ\exp(-\tilde{\mathbf{e}}_n^q) = \exp(\tilde{\mathbf{e}}_{n+1}^q)\circ\exp\left(-h\,\widetilde{\Delta\mathbf{q}}_n\right)\circ\exp(-\tilde{\mathbf{e}}_n^q),\end{aligned}$$

Author's personal copy

see Eq. (29). The resulting error recursion

$$\exp(\tilde{\mathbf{e}}^q_{n+1}) = \exp\left(\tilde{\mathbf{I}}^q_n\right)\circ\exp\left(-h\,\widetilde{\Delta\mathbf{q}}(t_n)\right)\circ\exp(\tilde{\mathbf{e}}^q_n)\circ\exp\left(h\,\widetilde{\Delta\mathbf{q}}_n\right) \tag{55}$$

for one time step of the Lie group integrator is nonlinear and may be transformed by recursive application of the Baker–Campbell–Hausdorff formula, see Eq. (35):

$$\begin{aligned}\exp(\tilde{\mathbf{e}}^q_{n+1}) &= \exp\left(\tilde{\mathbf{I}}^q_n\right)\circ\exp\left(-h\,\widetilde{\Delta\mathbf{q}}(t_n)\right)\circ\exp(\tilde{\mathbf{e}}^q_n)\circ\exp\left(h\,\widetilde{\Delta\mathbf{q}}(t_n)\right)\circ\exp\left(-h\,\widetilde{\Delta\mathbf{q}}(t_n)\right)\circ\exp\left(h\,\widetilde{\Delta\mathbf{q}}_n\right)\\ &= \exp\left(\tilde{\mathbf{I}}^q_n\right)\circ\exp\left(\tilde{\mathbf{e}}^q_n + h\left[\tilde{\mathbf{e}}^q_n, \tilde{\mathbf{v}}(t_n)\right] + \mathcal{O}\left(h^2\right)\left\|\tilde{\mathbf{e}}^q_n\right\|\right)\circ\exp\left(h\left(\widetilde{\Delta\mathbf{q}}_n - \widetilde{\Delta\mathbf{q}}(t_n)\right) + \mathcal{O}\left(h^2\right)\left\|\widetilde{\Delta\mathbf{q}}_n - \widetilde{\Delta\mathbf{q}}(t_n)\right\|\right)\\ &= \exp\left(\tilde{\mathbf{I}}^q_n + \tilde{\mathbf{e}}^q_n + h\left(\widetilde{\Delta\mathbf{q}}_n - \widetilde{\Delta\mathbf{q}}(t_n)\right) + h\,\tilde{\mathbf{r}}_n\right)\end{aligned} \tag{56}$$

with

$$\begin{aligned}h\,\tilde{\mathbf{r}}_n &= h\left[\tilde{\mathbf{e}}^q_n, \tilde{\mathbf{v}}(t_n)\right] + \mathcal{O}\left(h^2\right)\left\|\tilde{\mathbf{e}}^q_n\right\| + \mathcal{O}\left(h^2\right)\left\|\widetilde{\Delta\mathbf{q}}_n - \widetilde{\Delta\mathbf{q}}(t_n)\right\| + \mathcal{O}(h)\left\|\tilde{\mathbf{I}}^q_n\right\|\\ &= h\left[\tilde{\mathbf{e}}^q_n, \tilde{\mathbf{v}}(t_n)\right] + \mathcal{O}\left(h^2\right)\left(\left\|\tilde{\mathbf{e}}^q_n\right\| + \left\|\tilde{\mathbf{e}}^{\mathbf{v}}_n\right\| + h\left\|\tilde{\mathbf{e}}^{\mathbf{a}}_n\right\| + h\left\|\tilde{\mathbf{e}}^{\mathbf{a}}_{n+1}\right\|\right) + \mathcal{O}\left(h^4\right)\end{aligned} \tag{57}$$

since $\left\|\tilde{\mathbf{I}}^q_n\right\| = \mathcal{O}\left(h^3\right)$, $\left\|\tilde{\mathbf{e}}^q_n\right\| = \mathcal{O}(h)$ and $\left\|\widetilde{\Delta\mathbf{q}}_n - \widetilde{\Delta\mathbf{q}}(t_n)\right\| = \left\|\tilde{\mathbf{e}}^{\mathbf{v}}_n\right\| + \mathcal{O}(h)\left(\left\|\tilde{\mathbf{e}}^{\mathbf{a}}_n\right\| + \left\|\tilde{\mathbf{e}}^{\mathbf{a}}_{n+1}\right\|\right) = \mathcal{O}(h)$. Because of Eq. (56), the equivalent $\mathbf{e}^q_{n+1}\in\mathbb{R}^k$ to the global error $\tilde{\mathbf{e}}^q_{n+1}\in\mathfrak{g}$ at time t_{n+1} may be expressed as

$$\mathbf{e}^q_{n+1} = \mathbf{e}^q_n + h\left(\Delta\mathbf{q}_n - \Delta\mathbf{q}(t_n)\right) + \mathbf{I}^q_n + h\mathbf{r}_n. \tag{58}$$

The only (but important) difference between this estimate for the Lie group time integrator and the classical estimate (51) from ODE theory is the additional higher order error term $h\mathbf{r}_n$.

As in Eqs. (53) and (54), we obtain from Eq. (58) a three-term recursion for $\mathbf{e}^q_n$:

$$(1-\alpha_m)\mathbf{e}^q_{n+1} - (1-2\alpha_m)\mathbf{e}^q_n - \alpha_m\mathbf{e}^q_{n-1} \tag{59}$$

$$\begin{aligned}&= h\left((1-\alpha_m)\mathbf{e}^{\mathbf{v}}_n + \alpha_m\mathbf{e}^{\mathbf{v}}_{n-1}\right) + h^2(0.5-\beta)\left(\left(1-\alpha_f\right)\mathbf{e}^{\dot{\mathbf{v}}}_n + \alpha_f\mathbf{e}^{\dot{\mathbf{v}}}_{n-1}\right) + h^2\beta\left(\left(1-\alpha_f\right)\mathbf{e}^{\dot{\mathbf{v}}}_{n+1} + \alpha_f\mathbf{e}^{\dot{\mathbf{v}}}_n\right)\\ &\quad + (1-\alpha_m)\mathbf{I}^q_n + \alpha_m\mathbf{I}^q_{n-1} + h^2(1-\alpha_m)\left(\beta\mathbf{I}^{\mathbf{a}}_n + (0.5-\beta)\mathbf{I}^{\mathbf{a}}_{n-1}\right) + h\left((1-\alpha_m)\mathbf{r}_n + \alpha_m\mathbf{r}_{n-1}\right)\\ &= \mathcal{O}(h)\sum_{k=0}^{2}\left(\left\|\mathbf{e}^q_{n+k-1}\right\| + \left\|\mathbf{e}^{\mathbf{v}}_{n+k-1}\right\| + h\left\|\mathbf{e}^{\boldsymbol{\lambda}}_{n+k-1}\right\| + h^2\left\|\mathbf{e}^{\mathbf{a}}_{n+k-1}\right\|\right) + \mathcal{O}\left(h^3\right)\end{aligned} \tag{60}$$

since the global errors in $\dot{\mathbf{v}}$ and the higher order terms $h\mathbf{r}_n$, $h\mathbf{r}_{n-1}$ satisfy the estimates (49) and (57).

5.4. Error recursion for the Lagrange multipliers

It is a key idea of the convergence analysis for generalized-α methods applied to constrained systems to consider the error recursion in a time-discrete difference approximation $(\mathbf{e}^q_n)' := (\mathbf{e}^q_n - \mathbf{e}^q_{n-1})/h$ of the global error in $\dot{q}(t)$, see [18]. This error recursion is obtained from the difference of Eq. (59) for time step $t_n \to t_{n+1}$ and Eq. (59) for time step $t_{n-1} \to t_n$. There are three terms in the right hand side of this difference that require special attention: The first one is the weighted sum $(1-\alpha_m)(\mathbf{e}^{\mathbf{v}}_n - \mathbf{e}^{\mathbf{v}}_{n-1}) + \alpha_m(\mathbf{e}^{\mathbf{v}}_{n-1} - \mathbf{e}^{\mathbf{v}}_{n-2})$ that may be substituted by Eq. (53). At second, there are terms $(1-\alpha_m)(\mathbf{I}^q_n - \mathbf{I}^q_{n-1})/h$ and $\alpha_m(\mathbf{I}^q_{n-1} - \mathbf{I}^q_{n-2})/h$ approximating a scaled time derivative of local errors in q. Note, that $\mathbf{I}^q_n = \mathcal{O}(h^3)$, i.e. $\mathbf{I}^q_n / h = \mathcal{O}(h^2)$, but the difference quotients $(\mathbf{I}^q_n - \mathbf{I}^q_{n-1})/h$ are nevertheless of size $\mathcal{O}(h^3)$, see Eq. (47).

Finally, there are terms $(1-\alpha_m)(\mathbf{r}_n - \mathbf{r}_{n-1})$ and $\alpha_m(\mathbf{r}_{n-1} - \mathbf{r}_{n-2})$ that are bounded by $\mathcal{O}(h)\left(\left\|\tilde{\mathbf{e}}^q_{n-1}\right\| + \left\|\tilde{\mathbf{e}}^q_{n-2}\right\| + \left\|(\tilde{\mathbf{e}}^q_n)'\right\| + \left\|(\tilde{\mathbf{e}}^q_{n-1})'\right\|\right)$ and some higher order terms since

$$\left[\tilde{\mathbf{e}}^q_n, \tilde{\mathbf{v}}(t_n)\right] - \left[\tilde{\mathbf{e}}^q_{n-1}, \tilde{\mathbf{v}}(t_{n-1})\right] = h\left[(\tilde{\mathbf{e}}^q_n)', \tilde{\mathbf{v}}(t_n)\right] + \mathcal{O}(h)\left\|\tilde{\mathbf{e}}^q_{n-1}\right\|.$$

These estimates may be summarized to

$$(1-\alpha_m)(\mathbf{e}^q_{n+1})' - (1-2\alpha_m)(\mathbf{e}^q_n)' - \alpha_m(\mathbf{e}^q_{n-1})' \tag{61}$$

$$\begin{aligned}&= h\sum_{k=0}^{3} b'_k\mathbf{e}^{\dot{\mathbf{v}}}_{n+k-2} + \mathcal{O}(h)\sum_{k=0}^{3}\left(\left\|\mathbf{e}^q_{n+k-2}\right\| + \left\|\mathbf{e}^{\mathbf{v}}_{n+k-2}\right\| + \left\|(\mathbf{e}^q_{n+k-2})'\right\|\right)\\ &\quad + \mathcal{O}\left(h^2\right)\sum_{k=0}^{3}\left(\left\|\mathbf{e}^{\boldsymbol{\lambda}}_{n+k-2}\right\| + \left\|\mathbf{e}^{\mathbf{a}}_{n+k-2}\right\|\right) + \mathcal{O}\left(h^3\right)\end{aligned}$$

Author's personal copy

with coefficients b'_k being given in ([18], Eq. (19)). As in [18], an error recursion for the Lagrange multipliers $\boldsymbol{\lambda}$ is obtained by left multiplication of Eq. (61) with $\frac{1}{h}\left[\left(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\right)^{-1}\mathbf{B}\right]\left(q(t_n)\right)$, see Eq. (50):

$$\sum_{k=0}^{3} b'_k \mathbf{e}^{\boldsymbol{\lambda}}_{n+k-2} = \mathcal{O}(1)\left\|(1-\alpha_m)\mathbf{B}\left(q(t_n)\right)\frac{(\mathbf{e}^q_{n+1})' - (\mathbf{e}^q_n)'}{h} + \alpha_m \mathbf{B}\left(q(t_n)\right)\frac{(\mathbf{e}^q_n)' - (\mathbf{e}^q_{n-1})'}{h}\right\| + \mathcal{O}(1)\sum_{k=0}^{3}\left(\left\|\mathbf{e}^q_{n+k-2}\right\| + \left\|\mathbf{e}^{\mathbf{v}}_{n+k-2}\right\| + h\left\|\mathbf{e}^{\boldsymbol{\lambda}}_{n+k-2}\right\| + h\left\|\mathbf{e}^{\mathbf{a}}_{n+k-2}\right\|\right) + \mathcal{O}\left(h^2\right). \tag{62}$$

In the classical case, estimates for $\mathbf{B}\left(q(t_n)\right)\left(\left(\mathbf{e}^q_{n+k}\right)' - \left(\mathbf{e}^q_{n+k-1}\right)'\right)/h, (k=0,1)$, in terms of scaled constraint residuals $\left\|\Phi(q_{n+k-l})\right\|/h^2$ are obtained from a time-discrete difference approximation for the second time derivative of constraints (8), see Lemma 2 and Lemma 3 in [18]. The extension of these results to the Lie group setting of the present paper is technically rather complicated and has to be omitted to keep the presentation compact. As in the classical case, see ([18], Eq. (37)), we get finally a four-term recursion for the global errors $\mathbf{e}^{\boldsymbol{\lambda}}_n$:

$$\sum_{k=0}^{3} b'_k \mathbf{e}^{\boldsymbol{\lambda}}_{n+k-2} = \mathcal{O}\left(\frac{1}{h^2}\right)\sum_{k=0}^{3}\left\|\Phi(q_{n+k-2})\right\| + \mathcal{O}(1)\sum_{k=0}^{3}\left(\left\|\mathbf{e}^q_{n+k-2}\right\| + \left\|\mathbf{e}^{\mathbf{v}}_{n+k-2}\right\| + \left\|(\mathbf{e}^q_{n+k-2})'\right\| + h\left\|\mathbf{e}^{\boldsymbol{\lambda}}_{n+k-2}\right\| + h\left\|\mathbf{e}^{\mathbf{a}}_{n+k-2}\right\|\right) + \mathcal{O}\left(h^2\right). \tag{63}$$

Formally, the first term at the right hand side of Eq. (63) vanishes, see Eq. (28), but it illustrates the influence of round-off errors that can not be avoided in a practical implementation of the method.

In contrast to the convergence analysis in the classical DAE case [18], the global errors $\mathbf{e}^{\mathbf{a}}_n$ can not be completely eliminated from the three- and four-term recursions (60), (61) and (63) for $\mathbf{e}^q_n$, $(\mathbf{e}^q_n)'$ and $\mathbf{e}^{\boldsymbol{\lambda}}_n$ since the Baker–Campbell–Hausdorff formula applied to Eq. (55) results in additional higher order error terms $h\mathbf{r}_n$ that reflect the nonlinearity of Lie group integrators. It is an important detail of the convergence analysis that the coefficient of $\left\|\mathbf{e}^{\mathbf{a}}_{n+k-2}\right\|$ in the four-term recursion (63) may be bounded by a factor of size $\mathcal{O}(h)$ comparing the numerical solution $\mathbf{a}_n$ *not* with $\dot{\mathbf{v}}(t_n)$ but with $\dot{\mathbf{v}}(t_n + (\alpha_m - \alpha_f)h)$ to define the global error $\mathbf{e}^{\mathbf{a}}_n$, see Eq. (48).

5.5. Convergence

The three-term recursions (60), (54) and (61) for the global errors $\mathbf{e}^q_n$, $\mathbf{e}^{\mathbf{v}}_n$ and $(\mathbf{e}^q_n)'$ in the differential components and the two- and four-term recursions (52) and (63) for the global errors $\mathbf{e}^{\mathbf{a}}_n$ and $\mathbf{e}^{\boldsymbol{\lambda}}_n$ in the algebraic components may be summarized to a coupled four-term recursion for the global errors in all solution components, see ([18], Eqs. (39), (40)). The stability of such coupled error recursions was studied in great detail to prove convergence of linear multi step methods applied to higher index DAEs ([21], Section VII.3), see also the discussion of technical details in the application to generalized-α methods in Section 3.4 of [18].

For fixed time step sizes h, the generalized-α Lie group time integrator has global errors $\mathcal{O}(h^2)$ in all solution components (second order convergence), if the classical order condition $\gamma = 0.5 + \alpha_f - \alpha_m$ and the additional stability conditions

$$\alpha_m < \alpha_f < 0.5, \beta > 0.25 + \left(\alpha_f - \alpha_m\right)/2 \tag{64}$$

are satisfied and the errors in all initial values are of size $\mathcal{O}(h^2)$. The first order approximation $\mathbf{a}_0 = \mathbf{v}_0$ may introduce slightly larger errors in the initial phase that are rapidly damped out during integration [21]. In the numerical tests of Section 7, the order condition and the stability conditions (64) are always satisfied selecting parameters α_m, α_f, β, γ according to [15] with a spectral radius at infinity $\rho_\infty \in [0, 1)$.

6. Updated Lagrangian method

The updated Lagrangian method proposed in [2] for the integration of large rotation variables is also closely related to the strategy proposed by Munthe-Kaas for Lie group integrators. This method, which will be used in the following numerical examples as a reference solution, is briefly described here. The key idea is, at each time step, to map the differential equation to a local coordinate system. In the coordinate system, any classical integrator can be used to solve the dynamic problem. More precisely, at the current time t, the relative motion with respect to the previous time step t_n is parameterized using a vector $\mathbf{q}^n(t) \in \mathbb{R}^k$ and the exponential map according to

$$q(t) = q_n \circ \exp\left(\widetilde{\mathbf{q}^n(t)}\right). \tag{65}$$

Two successive differentiations of this equation yield

$$\mathbf{v}(t) = \mathbf{T}\left(\mathbf{q}^n(t)\right)\dot{\mathbf{q}}^n(t) \tag{66}$$

Author's personal copy

$$\dot{\mathbf{v}}(t) = \mathbf{T}\left(\mathbf{q}^n(t)\right)\ddot{\mathbf{q}}^n(t) + \dot{\mathbf{T}}\left(\mathbf{q}^n(t), \dot{\mathbf{q}}^n(t)\right)\dot{\mathbf{q}}^n(t). \tag{67}$$

The numerical solution at time t_n (resp. t_{n+1}) is denoted by $\mathbf{q}_0^n$ (resp. $\mathbf{q}_1^n$). Observing that $\mathbf{T}(0) = \mathbf{I}_k$ and $\dot{\mathbf{T}}(0, \dot{\mathbf{q}})\dot{\mathbf{q}} = 0_k$, we have

$$\mathbf{q}_0^n = 0 \tag{68}$$

$$\dot{\mathbf{q}}_0^n = \mathbf{v}_n \tag{69}$$

$$\ddot{\mathbf{q}}_0^n = \dot{\mathbf{v}}_n. \tag{70}$$

Using this strategy with the generalized-α method leads to the following algorithm

$$\mathbf{M}(q_{n+1})\dot{\mathbf{v}}_{n+1} = -\mathbf{g}(q_{n+1}, \mathbf{v}_{n+1}, t_{n+1}) - \mathbf{B}^T(q_{n+1})\boldsymbol{\lambda}_{n+1} \tag{71}$$

$$\Phi(q_{n+1}) = 0 \tag{72}$$

$$q_{n+1} = q_n \circ \exp\left(\widetilde{\mathbf{q}_1^n}\right) \tag{73}$$

$$\mathbf{v}_{n+1} = \mathbf{T}(\mathbf{q}_1^n)\dot{\mathbf{q}}_1^n \tag{74}$$

$$\dot{\mathbf{v}}_{n+1} = \mathbf{T}(\mathbf{q}_1^n)\ddot{\mathbf{q}}_1^n + \dot{\mathbf{T}}(\mathbf{q}_1^n, \dot{\mathbf{q}}_1^n)\dot{\mathbf{q}}_1^n \tag{75}$$

$$\mathbf{a}_{n+1} = \mathbf{T}(\mathbf{q}_1^n)\mathbf{a}_1^n + \dot{\mathbf{T}}(\mathbf{q}_1^n, \dot{\mathbf{q}}_1^n)\dot{\mathbf{q}}_1^n \tag{76}$$

$$\mathbf{q}_1^n = h\mathbf{v}_n + (0.5-\beta)h^2\mathbf{a}_n + \beta h^2\mathbf{a}_1^n \tag{77}$$

$$\dot{\mathbf{q}}_1^n = \mathbf{v}_n + (1-\gamma)h\mathbf{a}_n + \gamma h\mathbf{a}_1^n \tag{78}$$

$$(1-\alpha_m)\mathbf{a}_1^n + \alpha_m\mathbf{a}_n = \left(1-\alpha_f\right)\ddot{\mathbf{q}}_1^n + \alpha_f\dot{\mathbf{v}}_n. \tag{79}$$

This set of equations can be solved for given values of q_n, $\mathbf{v}_n$, $\dot{\mathbf{v}}_n$ and $\mathbf{a}_n$. In contrast with the previous scheme, the integration formulae are defined in the local coordinate system and are not based on purely geometric quantities as angular velocities and accelerations. Moreover, the complexity of the updated Lagrangian algorithm is increased due to the presence of velocity and acceleration transformation formulae (74–76). The main difference between the algorithm (71–79) and the approach in [2] comes from the definition of the acceleration-like variable, which is based on a linear combination at acceleration level here and on a linear combination at residual force level in [2].

Eq. (65) defines a local parameterization of the manifold based on the exponential map. Other updated Lagrangian methods can be obtained if other coordinate maps $\mu : \mathbb{R}^k \to G$ are used

$$q(t) = q_n \circ \mu\left(\mathbf{q}^n(t)\right). \tag{80}$$

Following the same idea as in [26], the examples below will be treated using an updated Lagrangian method with a parameterization based on the conformal rotation vector (CRV). If $G = SO(3)$, the CRV coordinate map is given by

$$\mu_{CRV,SO(3)}(\mathbf{c}) = \frac{1}{(4-c_0)^2}\left[\left(c_0^2 + 8c_0 - 16\right)\mathbf{I} + 2\mathbf{c}\mathbf{c}^T + 2c_0\tilde{\mathbf{c}}\right], \quad c_0 = \frac{16-\|\mathbf{c}\|^2}{8} \tag{81}$$

and the tangent operator is

$$\mathbf{T}_{CRV,SO(3)}(\mathbf{c}) = \frac{2}{(4-c_0)^2}\left(c_0\mathbf{I} + \frac{1}{4}\mathbf{c}\mathbf{c}^T - \tilde{\mathbf{c}}\right). \tag{82}$$

Sometimes, those coordinates are also referred to as the Wiener–Milenkovic parameters. The definition of the CRV coordinates is easily extended to any Lie group G formed by a multiple Cartesian product of $\mathbb{R}^3$ and $SO(3)$.

7. Numerical examples

In the following numerical examples, two time integration methods are compared

- The "Lie-α" method is the proposed generalized-α method for dynamic systems on Lie group, see Section 4.
- The "UL-CRV" method is the updated Lagrangian method based on the conformal rotation vector parameterization, see Section 6.

Author's personal copy

7.1. Heavy top — constrained formulation

As shown in Fig. 1, the heavy top is a rotating body fixed to the ground by a spherical joint. In principle, the motion could be described in $SO(3)$ if the equilibrium of momentum were expressed with respect to the fixed point. However, in order to test our algorithms for constrained systems, the translation of the center of mass and the rotation of the body are considered as independent variables. This strategy is in agreement with the finite element approach described in [1]. The configuration of the system is thus represented by $q = (\mathbf{x}, \mathbf{R}) \in \mathbb{R}^3 \times SO(3)$. However, due to the fixed point condition, the variables $\mathbf{x}$ and $\mathbf{R}$ have to satisfy three kinematic constraints so that the motion is restricted to a 3-dimensional submanifold of $\mathbb{R}^3 \times SO(3)$. The motion of the top is described by a set of differential-algebraic equations

$$m\ddot{\mathbf{x}} - \boldsymbol{\lambda} = m\boldsymbol{\gamma} \tag{83}$$

$$\mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \tilde{\mathbf{X}}\mathbf{R}^T\boldsymbol{\lambda} = \mathbf{0} \tag{84}$$

$$-\mathbf{x} + \mathbf{R}\mathbf{X} = \mathbf{0}. \tag{85}$$

The vector $\mathbf{x} \in \mathbb{R}^3$ represents the position of the center of mass in the inertial frame, $\mathbf{X} \in \mathbb{R}^3$ represents the constant position of the center of mass in the body-fixed frame and $\mathbf{R} \in SO(3)$ is the 3×3 rotation matrix of the body. m is the mass of the top and the inertia tensor $\mathbf{J}$ in the body-fixed frame is defined with respect to the center of mass. $\boldsymbol{\Omega} \in \mathbb{R}^3$ is the angular velocity in the body-fixed frame and $\boldsymbol{\gamma}$ is the 3×1 vector of gravity acceleration in the inertial frame. The third equation is a set of 3 algebraic constraints and $\boldsymbol{\lambda}$ is the associated 3×1 vector of Lagrange multipliers. The constraint gradient matrix is given by $\mathbf{B}(q) = \left[-\mathbf{I}_3 - \mathbf{R}\tilde{\mathbf{X}}\right]$.

In the numerical tests, the parameters of the model are defined as $m = 15$ kg, $\mathbf{J} = \mathrm{diag}(0.234375, 0.46875, 0.234375)\,\mathrm{kg\,m^2}$, $\mathbf{X} = [0.1.0.]^T$ m, $\boldsymbol{\gamma} = [0.0. - 9.81]^T\,\mathrm{m/s^2}$ and the initial conditions are $\mathbf{R}(0) = \mathbf{I}_3$, $\boldsymbol{\Omega}(0) = [0.150. - 4.61538]^T$ rad/s. The initial values $\dot{\mathbf{x}}(0)$, $\ddot{\mathbf{x}}(0)$ and $\dot{\boldsymbol{\Omega}}(0)$ are set to be consistent with the constraints and the equations of motion. In Fig. 2, the numerical results are obtained using a time step $h = 0.002$ s and the algorithmic parameters are defined according to the Chung–Hulbert scheme [15] in order to have a spectral radius at infinity $\rho_\infty = 0.9$. The solutions obtained by the proposed Lie group method and by the updated Lagrangian method are compared. The vertical displacement of the center of mass x_3 is shown in the interval [0, 2] s and the fast rotation of the top is represented by the second component of the Cartesian rotation vector ψ_2 in the interval [0, 0.2] s. The figure shows numerical oscillations for the multiplier $\boldsymbol{\lambda}_1$ at the beginning of the simulation, which are stabilized efficiently by the time integrator. The mean number of Newton iterations per time step is equal to 3 for both methods.

The energy is not exactly preserved but the drifts are much smaller for the Lie-α than for the UL-CRV method. Fig. 2 shows that total energy is continuously increasing for computations using UL-CRV algorithm. This behavior is due to the significant nonlinearity of the CRV parameterization. When performing the same computations using the rotational vector to parameterize rotations in the updated Lagrangian method, this effect does not appear and the energy evolution is much closer to the results presented by the Lie group integrator, as shown in [14].

In order to analyze the convergence, a reference solution was computed using a small time step $h = 1.5625\,\mathrm{e} - 5\,\mathrm{s}$. In agreement with the theoretical convergence analysis, second-order accuracy is observed both for differential and algebraic variables $\mathbf{x}$ and $\boldsymbol{\lambda}$. The errors in $\mathbf{x}$ are closely related with the errors in $\mathbf{R}$ since the constraint (85) is satisfied at each time step with a precision fixed by the tolerance of the Newton iteration process.

7.2. Right-angle cantilever beam

The second example is a classical benchmark for nonlinear dynamic beam formulations, see [26] and references therein. An L shape cantilever beam made of two 10-meter-long segments is set into motion by applying an out-of-plane concentrated load at its elbow as is depicted in Fig. 3. The applied force is linearly increasing on $0 \le t \le 1$ s to reach the maximum value of 50 N and then is decreasing on $1 \le t \le 2$ s to reach the 0 value. In the remaining time until $t = 30$ s, the cantilever is undergoing free vibrations of finite amplitude with combined bending and torsion.

The equations of motion are derived using a classical geometrically exact beam formulation described in [26]. The translation and rotational fields along the beam axis are discretized according to the finite element method. Compared to [26], the beam

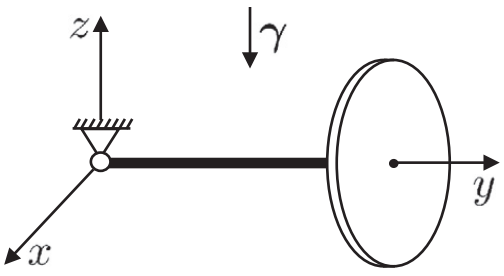

Fig. 1. Heavy top.

Author's personal copy

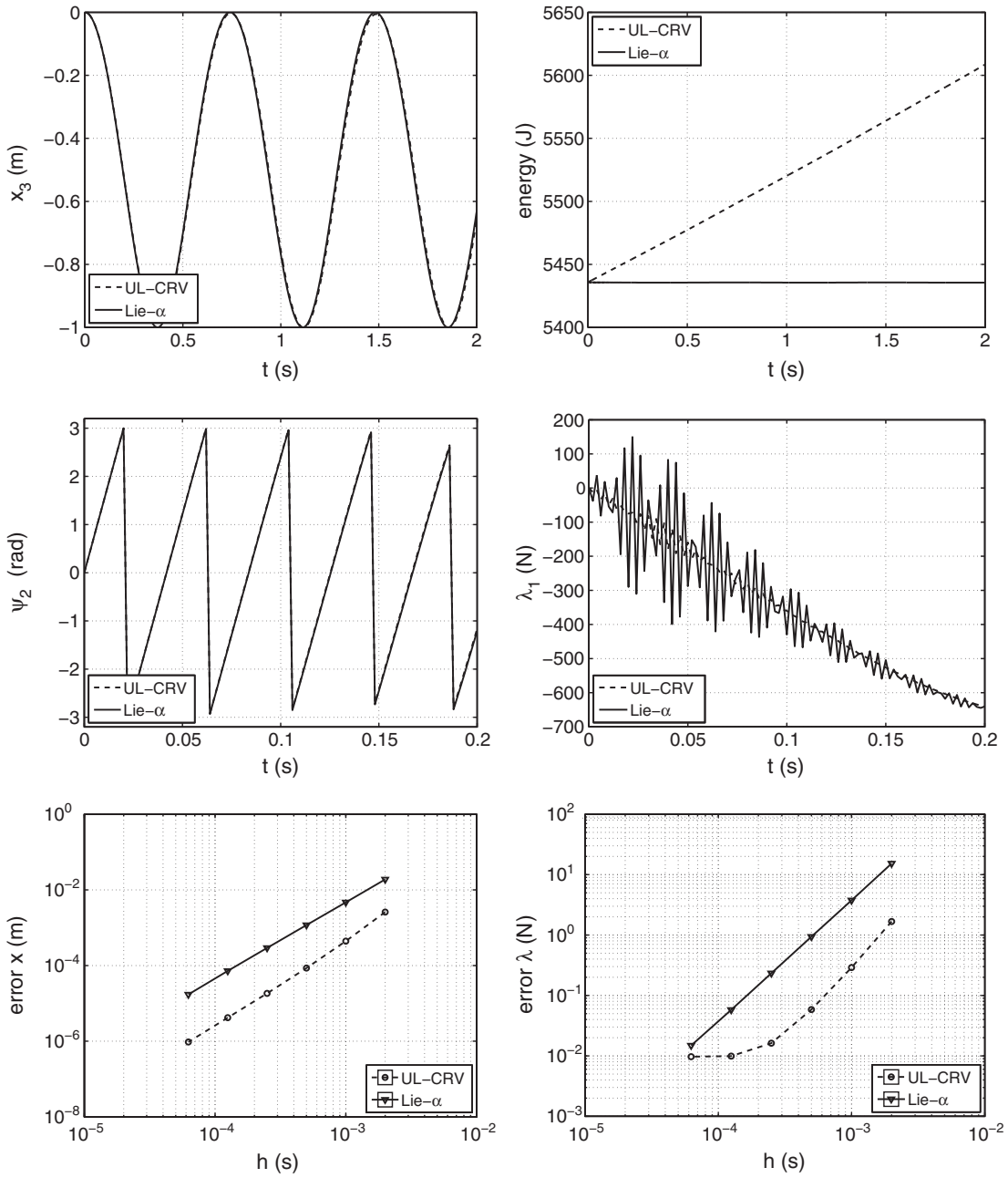

Fig. 2. Heavy top: Simulation results.

formulation is adapted for the Lie group solver, with the main difference that the equations of motion are now formulated in the Lie algebra and not in a parameter space. This significantly simplifies the formulation and the implementation of the beam model. The model does not include any kinematic constraint, which means that the equations of motion have the structure of an ODE.

The computation is carried out using 10 beam elements and a constant time step $h = 0.2$ s. The algorithmic parameters are defined according to Hilber–Hughes–Taylor method (HHT-method) with the numerical damping coefficient $\alpha = 0.05$. This choice leads to a spectral radius $\rho_\infty = 0.905$. The results are shown in Fig. 4. The response for out-of-plane displacements of the tip is plotted. Note that the amplitudes of the vibration have same order of magnitude as the structure dimensions, which means that the beam is really subjected to very large displacements and rotations in 3D. The results obtained using the proposed Lie group method and the updated Lagrangian approach are in close agreement with each other as well as with [26] and references therein. There is no significant difference in the mean number of Newton iterations per time step, which is equal to 3.77 for the UL-CRV algorithm and to 3.83 for the Lie-α algorithm. Even though the method was not designed to preserve exactly the energy, one can appreciate the small variations of energy during the free-vibration phase of the simulation.

Based on a reference solution computed using the same algorithm and a small time step $h = 7.8\text{e}-4$ s, a convergence study of displacements has been achieved. As predicted by the theoretical analysis, the proposed Lie group generalized-α method is second-order accurate and it has a similar level of accuracy as the updated Lagrangian method for computing the solution of nonlinear structural dynamics problems with large finite rotations. Indeed, the main advantage of the Lie group method is the

Author's personal copy

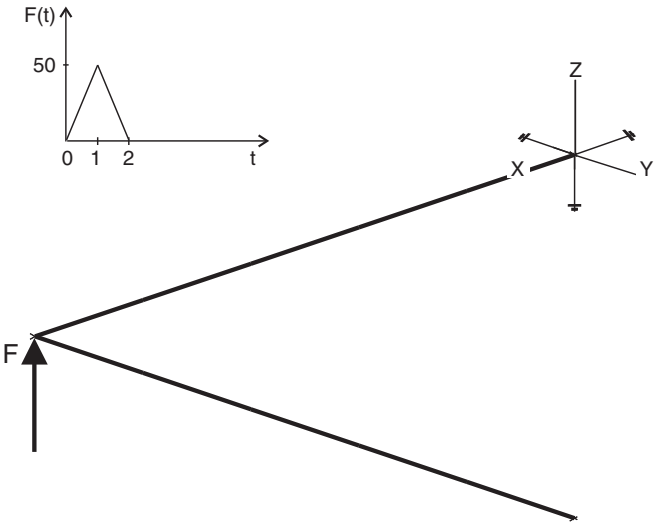

Fig. 3. Right-angle beam.

simpler formulation of the integrator and of the nonlinear beam element in the Lie algebra, compared to formulations in the parameter space.

7.3. Four-bar flexible mechanism with a misaligned hinge

This four-bar mechanism problem, which is illustrated in Fig. 5, was presented by Bauchau and Bottasso [27]. The system is composed of three flexible bars connected to each other and to the ground by ideal revolute joints. The axis of the joints at points *A*, *B* and *D* are normal to the plane of the mechanism but the axis of the joint at point *C* makes an angle of 5° with the *z* axis, which simulates a fabrication default of the system. If all the bars were rigid, no motion would be possible for this mechanism. Motion is

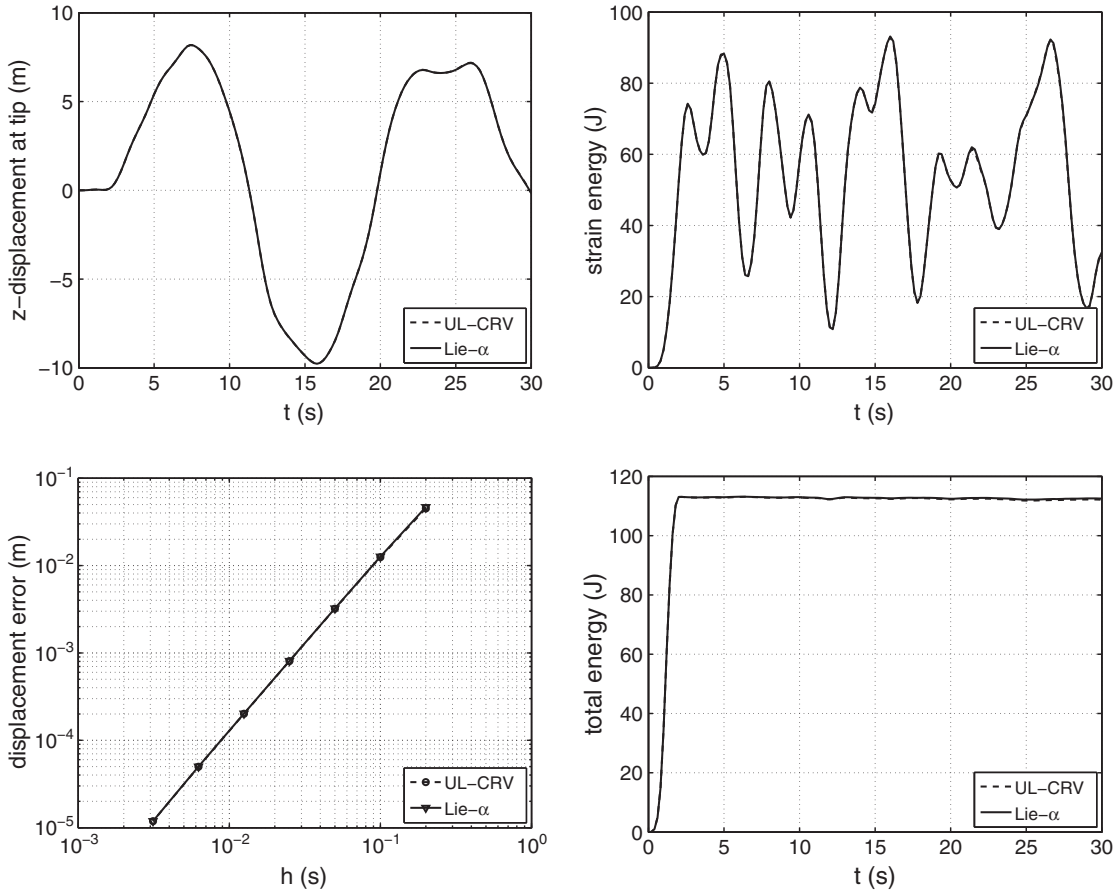

Fig. 4. Right-angle beam: Simulation results.

Author's personal copy

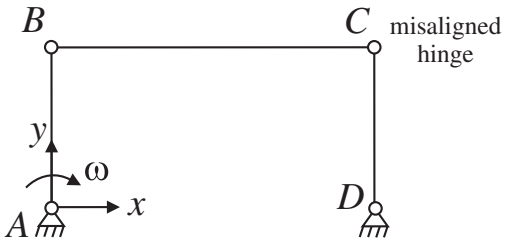

Fig. 5. Four-bar flexible mechanism.

only possible since the flexibility of the bars is taken into account. In this test-case, a constant rotation speed is imposed for the hinge at point A, which results in a cyclic 3D motion of the system. Indeed, deformations of the flexible bars have both in-plane and out-of-plane components. When bar AB rotates at constant speed, bar CD does not complete an entire turn but it oscillates back and forth.

It was not possible to have the whole set of parameters used in [27], which explains the fact that our simulation results are different. The parameters of the system are selected as follows. For the axis orientation of the defective hinge, the angle of 5° is measured positively about y-axis. For the three bars, the Young modulus and Poisson ratio are $E=2.1e11\,\mathrm{N/m^2}$, $\nu=0.3$, so that $G=E/(2(1+\nu))$. The other properties of the bars are given in Table 2. The relative angle θ_A of hinge A is imposed with a smooth acceleration phase between $t=t_0$ and $t=t_s$ followed by a constant speed phase according to the law

$$\theta_A(t)=\begin{cases}(\omega/t_s)\left(t^2/2+\left(t_s/(2\pi)\right)^2\left(\cos(2\pi t/t_s)-1\right)\right) & \text{for } t_0\le t\le t_s\\ \omega(t-t_s/2) & \text{for } t_s\le t\end{cases} \tag{86}$$

with $t_0=0\,\mathrm{s}$, $t_s=0.5\,\mathrm{s}$ and $\omega=-5\,\mathrm{rad/s}$.

The model of the flexible beams is based on the same nonlinear finite element formulation as in the previous example. Each bar is discretized into 10 finite elements. The step-size is selected as $h=2.\mathrm{e}-2\,\mathrm{s}$ and the numerical parameters of the proposed Lie group method and of the updated Lagrangian method are selected according to the Chung–Hulbert method [15], with a spectral radius at infinite frequency $\rho_\infty=0.9$. Numerical results are shown in Fig. 6. As in the previous examples, there is no significant difference in the mean number of Newton iterations per time step, which is equal to 3.06 for the UL-CRV algorithm and to 3.05 for the Lie-α algorithm. Again, even though the Lie group formulation of the beam element and of the time integrator are much simpler, the results obtained by the two algorithms are in close agreement with each other and second-order convergence is observed for both displacements and Lagrange multipliers. This example shows how the theoretical conclusions of this study are valid for constrained flexible multibody systems.

8. Conclusions

This paper studies a Lie group extension of the generalized-α time integrator for the simulation of flexible multibody systems with kinematic constraints. The method provides an elegant solution to the rotation parameterization problem, it does not suffer from parameterization singularities and it does not require the definition of redundant rotational coordinates. Moreover, it is general and applicable to any mechanical model with large rotation variables.

Second-order accuracy is demonstrated in the DAE case and is verified for three examples of rigid and flexible body systems. The Lie group integrator is compared with a more classical updated Lagrangian method, which is also formulated in a Lie group setting. We conclude that the new method can compete with the more classical ones from the viewpoint of accuracy, stability and energy conservation.

The main advantages of the Lie group method are related with its particular formulation, which is both generic and remarkably simple compared to parameterization-based algorithms. Important simplifications appear in the expression of the iteration matrix involved in the Newton procedure to solve the nonlinear problem at each time step. Therefore, the implementation of the exact

Table 2
Parameters of the four-bar system. Parameters EI, GA_{shear} and I_i have the same values along the two principle axes of the cross section.

Bar	*AB*	*BC*	*CD*
Length (m)	0.12	0.24	0.12
Traction stiffness EA (N)	4.e7	4.e6	4.e5
Flexion stiffness EI ($\mathrm{Nm^2}$)	2.4e6	2.8e5	2.4e4
Torsion stiffness GJ ($\mathrm{Nm^2}$)	2.8e5	2.8e4	2.8e4
Shearing stiffness GA_{shear} (N)	2.e6	2.e5	2.e5
Mass per unit length (kg/m)	3.2	1.6	1.6
Section inertia about beam axis J_i (kgm)	2.4e−2	1.2e−2	1.2e−2
Section inertia about transverse axis I_i (kgm)	1.2e−2	6.e−3	6.e−3

Author's personal copy

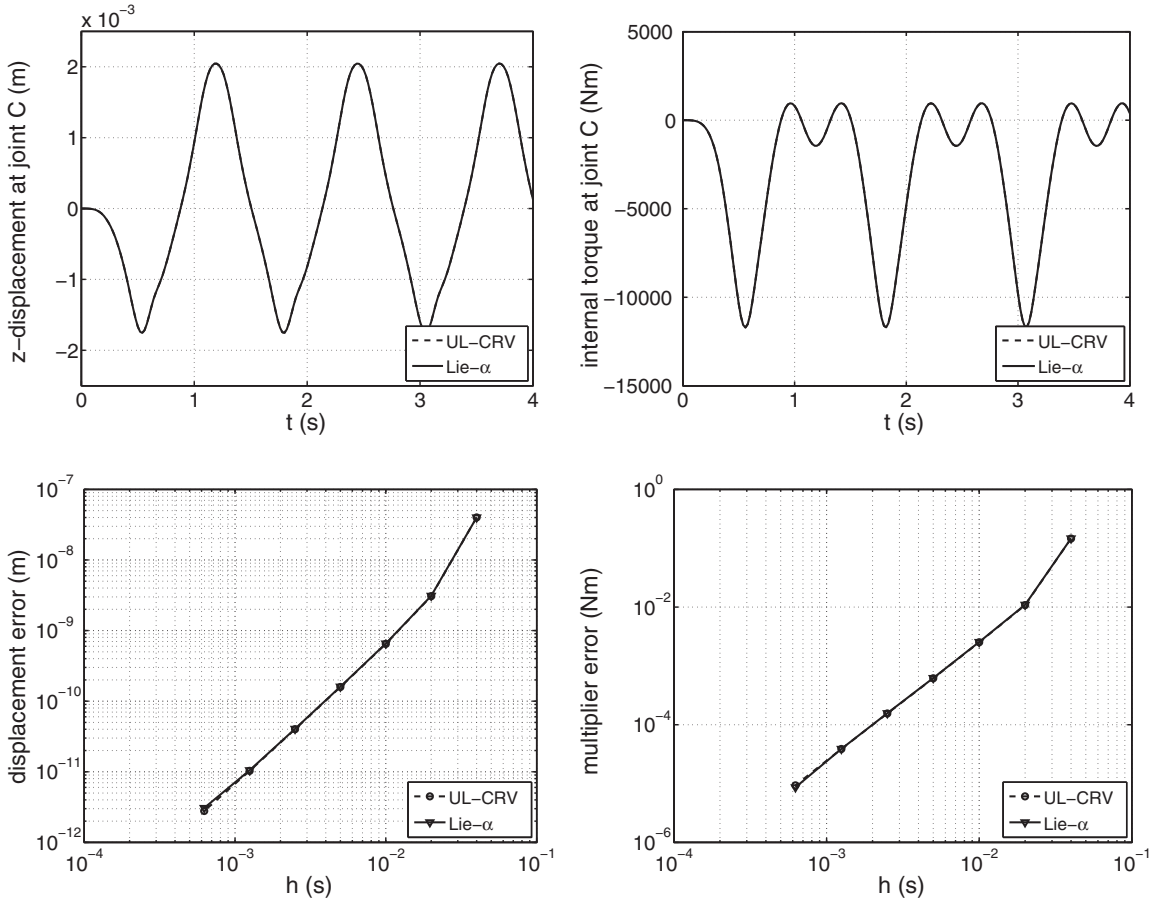

Fig. 6. Four-bar flexible mechanism: Simulation results.

expression of this matrix in a simulation code is more easily feasible in order to guarantee the fastest convergence rate of the iterations. The potential improvement in computational efficiency is an advantage for real-time applications, large scale problems, parametric studies and optimization. The simplicity of the formulation is also interesting for a semi-analytical sensitivity analysis or for model-based control schemes. Hence, Lie group time integrators are promising candidates for the development of robust, efficient and open simulation software for the analysis of flexible multibody systems.

References

[1] M. Géradin, A. Cardona, Flexible Multibody Dynamics: A Finite Element Approach, John Wiley & Sons, New York, 2001.
[2] A. Cardona, M. Géradin, Time integration of the equations of motion in mechanism analysis, Computers and Structures 33 (1989) 801–820.
[3] H. Lang, J. Linn, M. Arnold, Multibody dynamics simulation of geometrically exact Cosserat rods, Multibody System Dynamics 25 (2011) 285–312.
[4] P. Betsch, P. Steinmann, Constrained integration of rigid body dynamics, Computer Methods in Applied Mechanics and Engineering 191 (2001) 467–488.
[5] P. Crouch, R. Grossman, Numerical integration of ordinary differential equations on manifolds, Journal of Nonlinear Science 3 (1993) 1–33.
[6] H. Munthe-Kaas, Lie-Butcher theory for Runge–Kutta methods, BIT 35 (1995) 572–587.
[7] H. Munthe-Kaas, Runge–Kutta methods on Lie groups, BIT 38 (1998) 92–111.
[8] E. Celledoni, B. Owren, Lie group methods for rigid body dynamics and time integration on manifolds, Computer Methods in Applied Mechanics and Engineering 192 (3–4) (2003) 421–438.
[9] C. Bottasso, M. Borri, Integrating finite rotations, Computer Methods in Applied Mechanics and Engineering 164 (1998) 307–331.
[10] J. Simo, L. Vu-Quoc, On the dynamics in space of rods undergoing large motions — a geometrically exact approach, Computer Methods in Applied Mechanics and Engineering 66 (1988) 125–161.
[11] J. Simo, K. Wong, Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum, International Journal for Numerical Methods in Engineering 31 (1991) 19–52.
[12] A. Cardona, M. Géradin, A beam finite element non-linear theory with finite rotations, International Journal for Numerical Methods in Engineering 26 (1988) 2403–2438.
[13] A. Cardona, An integrated approach to mechanism analysis, Ph.D. thesis (1989).
[14] O. Brüls, A. Cardona, On the use of Lie group time integrators in multibody dynamics, ASME Journal of Computational and Nonlinear Dynamics 5 (3) (2010) 031002.
[15] J. Chung, G. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, ASME Journal of Applied Mechanics 60 (1993) 371–375.
[16] W. Boothby, An Introduction to Differentiable Manifolds and Riemannian Geometry, 2nd Edition, Academic Press, 2003.
[17] A. Iserles, H. Munthe-Kaas, S. Nørsett, A. Zanna, Lie-group methods, Acta Numerica 9 (2000) 215–365.
[18] M. Arnold, O. Brüls, Convergence of the generalized-α scheme for constrained mechanical systems, Multibody System Dynamics 18 (2) (2007) 185–202.
[19] A. Cardona, M. Géradin, Numerical integration of second order differential-algebraic systems in flexible mechanism dynamics, in: J.A.C. Ambrosio, M.F. Seabra Pereira (Eds.), Computer Aided Analysis of Rigid and Flexible Mechanical Systems, Vol. 268, Kluwer Academic Publishers, Dordrecht, 1994, pp. 501–529.

Author's personal copy

[20] C. Bottasso, O. Bauchau, A. Cardona, Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations, SIAM Journal on Scientific Computing 29 (1) (2007) 397–414.
[21] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II — Stiff and Differential-Algebraic Problems, 2nd Edition Springer-Verlag, Berlin Heidelberg New-York, 1996.
[22] J. Wensch, Extrapolation methods in Lie groups, Numerische Mathematik 89 (2001) 591–604.
[23] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations, Springer-Verlag, Berlin Heidelberg New-York, 2002.
[24] A. Müller, Approximation of finite rigid body motions from velocity fields, Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM) 90 (6) (2010) 514–521.
[25] S. Erlicher, L. Bonaventura, O. Bursi, The analysis of the generalized-α method for non-linear dynamic problems, Computational Mechanics 28 (2002) 83–104.
[26] E. Lens, A. Cardona, A nonlinear beam element formulation in the framework of an energy preserving time integration scheme for constrained multibody systems dynamics, Computers and Structures 86 (2008) 47–63.
[27] O. Bauchau, C. Bottasso, On the design of energy preserving and decaying schemes for flexible nonlinear multi-body systems, Computer Methods in Applied Mechanics and Engineering 169 (1999) 61–79.