MULTIBODY DYNAMICS 2011, ECCOMAS Thematic Conference
J.C. Samin, P. Fisette (eds.)
Brussels, Belgium, 4-7 July 2011

LIE-GROUP INTEGRATION METHOD FOR CONSTRAINED MULTIBODY SYSTEMS

Zdravko Terze[*], Dario Zlatar[*] and Andreas Müller[†]

[*]Department of Aeronautical Engineering
Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Ivana Lučića 5,
10000 Zagreb, Croatia
e-mail: zdravko.terze@fsb.hr, dario.zlatar@fsb.hr
web page: http://www.fsb.hr/aero/zterze.htm

[†] Chair of Mechanics and Robotics
University Duisburg-Essen, Lotharstrasse 1, 47057 Duisburg, Germany
e-mails: andreas-mueller@uni-due.de

Keywords: Lie-groups, Multibody Systems Dynamics, Numerical Integration Methods, DAE systems, Constraint Violation Stabilization, Manifolds, Munthe-Kaas method

Abstract. *A Lie-group integration method for constrained multibody systems is proposed in the paper. Mathematical model of multibody system dynamics is shaped as DAE system of equations of index 1, while dynamics is evolving on Lie-group introduced as system 'state space formulation'. Integration algorithm operates directly with angular velocities and rotational matrices and no local (generalized) coordinates are introduced. The basis of the method is Munthe-Kaas algorithm for ODE on Lie-groups, which is re-formulated and expanded to be applicable for the integration of constrained multibody dynamics, where constraint violation stabilization is one of the important issues that must be successfully solved.*

1 INTRODUCTION

Design of the integration methods for dynamic simulation of mechanical systems based on Lie-group formulations offers some attractive features such as simplicity and numerical efficiency of the code. The possibility of using global rotational coordinates for kinematic definition of the system certainly appeals for numerous applications. The additional motivation for taking this approach can be based on the fact that geometric formulation provides integration schemes that can preserve global characteristics of motion (such as conservation of the first integrals) in 'more naturally' manner than 'classical' methods that operate in vector spaces.

2 GEOMETRIC DAE INTEGRATION PROCEDURE FOR MBS

2.1 Lie-group ODE integrator

The configuration of a rotating rigid body is given by a rotation matrix $\mathbf{R}$ that belongs to Special Orthogonal Group $SO(3)$:

$$SO(3) = \left\{ \mathbf{R} \in \mathcal{R}^{3x3} : \mathbf{R}\mathbf{R}^T = \boldsymbol{I}, \det \mathbf{R} = +1 \right\} \tag{1}$$

From the geometrical point of view, $SO(3)$ can be considered as a differential manifold (space with different possibilities of parametarizations on which we can do calculus). Tangent vectors $\dot{\mathbf{R}}$ to $SO(3)$ at point $\mathbf{R} \in SO(3)$ of the manifold belong to the tangent space $T_{\mathbf{R}}SO(3)$, $\dot{\mathbf{R}} \in T_{\mathbf{R}}SO(3)$. Moreover, $SO(3)$ has the properties of Lie-group, where the tangent space at the group identity $\boldsymbol{I}$ has an additional structure. This vector space is equipped with matrix commutator and constitutes Lie-algebra of $SO(3)$, the set of skew-symmetric matrices denoted by $so(3)$. The element of Lie-algebra $\tilde{\boldsymbol{\omega}} \in so(3)$ can be identified with $\mathcal{R}^3$ *via* mapping operator which maps a vector $\boldsymbol{\omega} \in \mathcal{R}^3$ to a matrix $\tilde{\boldsymbol{\omega}} \in so(3)$. Between the elements of the group $\mathbf{R} \in SO(3)$ and Lie-algebra $\tilde{\boldsymbol{\omega}} \in so(3)$ exists natural correspondence *via* exponential map [1, 2, 11] and the expression $\mathbf{R}(t) = \exp(t\tilde{\boldsymbol{\omega}})$ is solution of the inital value problem

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\tilde{\boldsymbol{\omega}}, \; \mathbf{R}(0) = \boldsymbol{I} \, . \tag{2}$$

Also, the Lie-algebra element $\tilde{\boldsymbol{\omega}}$ defines left-invariant vector field $\mathbf{X}_{\tilde{\boldsymbol{\omega}}}$ on $SO(3)$ *via* relation $\mathbf{X}_{\tilde{\boldsymbol{\omega}}}(\mathbf{R}) = L'_{\mathbf{R}}(\tilde{\boldsymbol{\omega}})$, $\mathbf{X}_{\tilde{\boldsymbol{\omega}}}(\mathbf{R}) \in T_{\mathbf{R}}SO(3)$, where the tangent map $L'_{\mathbf{R}}(\tilde{\boldsymbol{\omega}}) : TSO(3) \to TSO(3)$ is given as $L'_{\mathbf{R}}(\tilde{\boldsymbol{\omega}}) = \mathbf{R}\tilde{\boldsymbol{\omega}}$ defining left kinematic Poisson relation $\dot{\mathbf{R}} = \mathbf{R}\tilde{\boldsymbol{\omega}}$. Equivalently, instead of using body coordinates $\tilde{\boldsymbol{\omega}}$, the right Poisson equation can be formulated as $\dot{\mathbf{R}} = \tilde{\boldsymbol{\omega}}_S\mathbf{R}$ by using angular velocity expressed *via* spatial coordinates $\tilde{\boldsymbol{\omega}}_S$ [3]. As explained above, left and right Poisson kinematic equation represent differential equation on $SO(3)$, since $\mathbf{R} \in SO(3)$ and the angular velocity tensor ($\tilde{\boldsymbol{\omega}}$ or $\tilde{\boldsymbol{\omega}}_S$) belongs to Lie-algebra $so(3)$. In the case when angular velocity is not a constant skew-symmetric matrix (as it is the case in the equation (2), which has solution $\mathbf{R}(t) = \exp(t\tilde{\boldsymbol{\omega}})$), body kinematics that evolves on $SO(3)$ is to be computed numerically. In the general setting, the objective is to find solution of differential equation on a matrix Lie-group G, with Lie-algebra $\boldsymbol{g}$, in the form

$$\dot{Y}(t) = A(Y(t))Y(t) \, , \tag{3}$$

where $Y(0) \in G$ and $A(Y) \in g$ for all $Y \in G$ and right trivialization is used. The equation (3) can be numerically solved by using different geometric integration methods [1, 2], such as

Munthe-Kaas algorithm [2, 7] that assumes result in the form

$$Y(t) = \exp(u(t))Y_0\,, \tag{4}$$

where $u(t)$ is the solution of

$$\dot{u} = d\exp_u^{-1}(A(Y(t)))\,,\; u(0) = 0\,. \tag{5}$$

By following this route, the numerical solution of kinematic equation (3) can be incorporated into the computational procedures based on Newton-Euler formulation, where rotational dynamics of rigid body is studied directly on $SO(3)$. This leads to more efficient procedures since no local parametarisation of 3D rotation is needed.

However, for a more general use of geometric integration algorithms within the framework of multibody system (MBS) dynamics, an efficient and accurate method of treating of kinematical constraints is needed. This issue is a central topic within MBS dynamics, and computational procedures based on the 'straightforward' Lie-group ODE integrators can not provide a sufficient framework.

2.2 Lie-group DAE integration procedure

2.2.1 Procedure framework

The objective of the paper is to describe an integration method that operates on a Lie-group and is suitable for dynamic simulation of constrained MBS. The method is based on Munthe-Kaas algorithm for ODEs on Lie-groups, which will be extended with the sub-routines for treating kinematical constraints i.e. the algorithm is designed to integrate differential-algebraic (DAE) equations of constrained MBS dynamics.

The configuration space of MBS is modeled as $G = \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times SO(3) \times ... \times SO(3)$, which is a n-dimensional manifold with Lie-group properties, consisting of translational and rotational kinematical domains of each rigid body in MBS. The Lie-group composition operation $G \times G \to G$ is introduced by $p_{com} = p_1 \circ p_2$, where $p_1, p_2, p_{com} \in G$ and the identity element e of the group is defined as $p \circ e = e \circ p = p, \forall p \in G$. The Lie-algebra $\mathfrak{g} = T_e G$ (vector space that is isomorphic to $\mathcal{R}^n$) is defined as the tangent space $T_p G$ at the identity $p{=}e$. The tangent vector in $T_p G$ (at any point $p \in G$) can be represented in Lie-algebra $\mathfrak{g}$ *via* derivation L'_p of the left translation map $L_p : G \to G, y \mapsto p \circ y$. Thus, for $y{=}e$, we can define bijection $L'_p(e) : \mathfrak{g} \to T_p G, \tilde{\mathbf{\Omega}} \mapsto L'_p(e) \cdot \tilde{\mathbf{\Omega}}$, where $L'_p(e) \cdot \tilde{\mathbf{\Omega}}$ is directional derivative of L_p at the point $y{=}e$ in direction of $\tilde{\mathbf{\Omega}} \in \mathfrak{g}$ (since G is Lie-group, the element of Lie-algebra $\tilde{\mathbf{\Omega}}$ defines left invariant vector field on G, similarly as it was the case with $SO(3)$).

To incorporate kinematical constraints of MBS, the function $\mathbf{\Phi} : G \to \mathcal{R}^m$ are imposed on G, meaning that system is constrained to evolve on the n-m dimensional sub-manifold $S = \{p \in G : \mathbf{\Phi}(p) = 0\}$. Consequently, dynamic equations of MBS are shaped in the form [4]

$$\begin{gathered} \mathbf{M}(p)\dot{\mathbf{v}} + \mathbf{Q}(p, \mathbf{v}, t) + \mathbf{C}^T(p)\boldsymbol{\lambda} = 0 \\ \mathbf{\Phi}(p) = 0 \\ \dot{p} = L'_p(e) \cdot \tilde{\mathbf{v}}\,, \end{gathered} \tag{6}$$

where $\mathbf{M}$ is $n \times n$ dimensional inertia matrix, $\mathbf{v} \in \mathcal{R}^n$, $\mathbf{v} = [\mathbf{v}_1, ..., \mathbf{v}_k, \boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_k]^T$ are system velocities (k bodies are assumed), $\mathbf{Q}$ represents external and non-linear velocity forces, $\boldsymbol{\lambda} \in \mathcal{R}^m$ is the vector of Lagrangian multipliers and $\mathbf{C}$ is $m \times n$ dimensional constraint gradient matrix, such that $\boldsymbol{\Phi}'(p) \cdot \tilde{\boldsymbol{\Omega}} = \mathbf{C}(p)\boldsymbol{\Omega}, \forall \boldsymbol{\Omega} \in \mathcal{R}^n$ is valid. The equation (6) represents DAE system of index 3. Within the framework of the proposed integration procedure, the equation (6) will be re-shaped into the DAE of index 1 form by including kinematical constraints at the acceleration level $\ddot{\boldsymbol{\Phi}}(p, \mathbf{v}, \dot{\mathbf{v}}) = 0$ (instead of $\boldsymbol{\Phi}(p) = 0$) and integrated by the routine based on the Munthe-Kaas algorithm.

To ensure that kinematical constraints are satisfied during integration, constraint violation of the system velocities $\mathbf{v} \in \mathcal{R}^n$ and generalized positions $p \in G$ will be corrected by using stabilization algorithm described in [6].

2.2.2 Integration algorithm

The configuration space of MBS at the generalized position level is modeled as Lie-group $G = \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times SO(3) \times ... \times SO(3)$ where translation and rotation of each rigid body is included in the n-dimensional configuration domain and element of the group is given as $p = (\mathbf{x}_1, ..., \mathbf{x}_k, \mathbf{R}_1, ..., \mathbf{R}_k)$. However, for the purpose of the proposed time integration routine that operates 'simultaneously' at the generalized positions and generalized velocities, a system has to be modeled on the *2n*-dimensional Lie-group (a system 'state space') $TG = \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times SO(3) \times ... \times SO(3) \times ... \times \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times so(3) \times ... \times so(3)$ with the element

$$q = (\mathbf{x}_1, ..., \mathbf{x}_k, \mathbf{R}_1, ..., \mathbf{R}_k, \mathbf{v}_1, ..., \mathbf{v}_k, \tilde{\boldsymbol{\omega}}_1, ..., \tilde{\boldsymbol{\omega}}_k), \qquad (7)$$

and its Lie-algebra $TTG = \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times so(3) \times ... \times so(3) \times ... \times \mathcal{R}^3 \times ... \times \mathcal{R}^3 \times so(3) \times ... \times so(3)$ with the element given as

$$z = (\mathbf{v}_1, ..., \mathbf{v}_k, \tilde{\boldsymbol{\omega}}_1, ..., \tilde{\boldsymbol{\omega}}_k, \dot{\mathbf{v}}_1, ..., \dot{\mathbf{v}}_k, \dot{\tilde{\boldsymbol{\omega}}}_1, ..., \dot{\tilde{\boldsymbol{\omega}}}_k). \qquad (8)$$

By confining ourselves on a single body system to keep formulation short, we introduce operations in Lie-group TG and its Lie-algebra TTG as follows.

Product in TG: $(a, b, c, d) \cdot (e, f, g, h) = (a + e, b \cdot f, c + g, d + h)$.

Addition in TTG: $(v, w, c, d) + (\bar{v}, \bar{w}, \bar{c}, \bar{d}) = (v + \bar{v}, w + \bar{w}, c + \bar{c}, d + \bar{d})$.

Multiplication by scalar in TTG: $\alpha(v, w, c, d) = (\alpha v, \alpha w, \alpha c, \alpha d)$.

Exponential map in TTG: $\exp(v, w, c, d) = (v, \exp(w), c, d)$.

Here, on the right hand side of definitions, '$\cdot$' is the multiplication in $SO(3)$, '+' is addition in $\mathcal{R}^3$ and $so(3)$ and exp is exponential map on $so(3)$. The operations in TG and TTG of multibody system consisting of system of k bodies is defined component-wise equivalently as for a single body system.

With all Lie-group operation in place, the differential equation describing dynamics of MBS on Lie group TG can be written in the form

$$\dot{q} = F(q)q, \qquad (9)$$

where $q \in TG$ and $F : TG \rightarrow TTG$ is given by $F : q \rightarrow z$, where elements q and z are given by (7) and (8). During evaluation of $F : q \rightarrow z$, the variables $\mathbf{v} = \left[\dot{\mathbf{v}}_1, ..., \dot{\mathbf{v}}_k, \dot{\tilde{\boldsymbol{\omega}}}_1, ..., \dot{\tilde{\boldsymbol{\omega}}}_k\right]^T$ are determined by the system dynamics equation

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\xi} \end{bmatrix}, \tag{10}$$

which has to be solved (linear algebraic system for variables $\dot{\mathbf{v}}$ and $\boldsymbol{\lambda}$) within integration algorithm of the differential equation (9).

The equation (10) represents first two equations of system (6), shaped as DAE of index 1, where the acceleration kinematical constraints $\ddot{\mathbf{\Phi}}(p, \mathbf{v}, \dot{\mathbf{v}}) = 0$ are introduced as $\mathbf{C}\dot{\mathbf{v}} = \boldsymbol{\xi}$ [5].

The differential equation (9) of the system dynamics on TG has the same form as differential equation (3) and can be solved by using Munthe-Kaas (MK) type of integration algorithm [7]. Similarly as it was the case with (3), (9) can be solved by introducing local integration coordinate u in TTG that satisfy

$$\dot{u} = d\exp_u^{-1}(F(q)),\ u(0) = 0. \tag{11}$$

Within MK method, (9) is integrated in TTG and numerical solution is than pulled-back on TG via exponential mapping. The algorithm itself can be given in the form [8]

$q_0 = q_{w-1}$
for $i = 1, 2, \ldots, s$
$\quad u_i = h \sum_{j=1}^{s} a_{ij} \tilde{k}_j$
$\quad k_i = F(c_i h, \exp(u_i, q_0))$
$\quad \tilde{k}_i = \mathrm{dexpinv}(u_i, k_i, n)$
end
$v = h \sum_{j=1}^{s} b_j \tilde{k}_j$
$q_w = \exp(v, q_0)$

where the coefficients are given by the nth order Runge-Kutta method's Butcher table [2, 12] and function dexpinv is defined as follows [2, 8]

$$\mathrm{dexpinv}(u, k, n) = k - \frac{1}{2}[u, k] + \sum_{p=2}^{n-1} \frac{B_p}{p!} \overbrace{[u, [u, [\ldots, [u, k]]]}^{p}.$$

The variables $u_i, k_i, \tilde{k}_i$ are MK method internal integration variables [7, 8] (similar as those that are defined within the framework of RK algorithms - MK methods reduce to RK algorithms when operate in vector space), which have the same format as z given by (8), see also (11).

A numerical solution obtained by the described algorithm will satisfy constraint equation at the acceleration level $\ddot{\mathbf{\Phi}}(p, \mathbf{v}, \dot{\mathbf{v}}) = 0$ automatically, since this equation is directly incorporated in the function evaluation $F : TG \rightarrow TTG$ *via* formulation (10). However, the constraint equations for the generalized positions $\mathbf{\Phi}(p) = 0$ and velocities $\dot{\mathbf{\Phi}}(p, \mathbf{v}) = 0$ (which can be also written in the form $\mathbf{C}(p)\mathbf{v} = \boldsymbol{\xi}$ [5]) that are also part of DAE formulation (6), will be unavoidably violated during the straightforward integration based on the MK type of algorithm. To solve this problem, the constraint violation stabilization procedure at the position and velocity level has to be incorporated into the integration procedure by expanding the algorithm described above.

For this purpose we propose the projective stabilization method, that is based on nonlinear constrained least square problem given in the form

$$\min_{(p_k, \mathbf{v}_k)} \left\| \begin{pmatrix} p_k - \hat{p}_k \\ \mathbf{v}_k - \hat{\mathbf{v}}_k \end{pmatrix} \right\|_{\mathbf{w}}^2, \ \mathbf{\Phi}(p) = 0, \ \mathbf{R}_i \mathbf{R}_i^T = \boldsymbol{I}, \ \dot{\mathbf{\Phi}}(p, \mathbf{v}) = 0, \tag{12}$$

(where $\| \ \|_{\mathbf{w}}$ denotes the weighted norm). The projected variables have to satisfy constraint equations $\mathbf{\Phi}(p) = 0$, $\mathbf{R}_i \mathbf{R}_i^T = \boldsymbol{I}$ and $\dot{\mathbf{\Phi}}(p, \mathbf{v}) = 0$ to obtain stabilized values $p_k, \mathbf{v}_k$, as expressed in (12), while $\hat{p}_k, \hat{\mathbf{v}}_k$ are 'unstabilized' values that are obtained from the integrator for the current integration step. Here, it should be emphasized that equations $\mathbf{R}_i \mathbf{R}_i^T = \boldsymbol{I}$ are included in the projection algorithm to make sure that stabilization procedure does not undermine orthogonality of $\mathbf{R}_i$ that will be imanently preserved by the MK type of integration algorithm (numerical integration on Lie-group TG that include and preserve $SO(3)$ manifold of each rigid body rotation domain). Actually, after each integration step on Lie group TG, we adjust integration values to be in compliance with kinematical constraints $\mathbf{\Phi}(p) = 0$ and $\dot{\mathbf{\Phi}}(p, \mathbf{v}) = 0$, by preserving orthogonality of $\mathbf{R}_i$ during the process (we treat $\mathbf{R}_i$ as it is valid $\mathbf{R}_i \in GL(3)$ and impose orthogonality equations $\mathbf{R}_i \mathbf{R}_i^T = \boldsymbol{I}$ as external conditions during stabilization given by (12)).

Technically, the numerical solution of the projection step can be computed iteratively by using Gauss-Newton algorithm, which is essentially based on generalized inverses (or pseudo-inverse) of the system constraint matrix and represents well-known common procedure in domain of numerical solving of algebraic systems [6, 9]. The alternative method of stabilizing constraint violation that might be specially convenient to be used in Lie group setting is discussed in [5,10]. It is reported in [5] that both methods provide excellent and very comparable stabilization results.

2.3 Numerical example: heavy top

As a numerical illustration, the example of heavy top (that became as a sort of a benchmark problem for this kind of analysis) is included. Since heavy top is formulated as a constrained mechanical system that leads to DAE formulation (DAE of index 1 in this case), the equations that govern system dynamics and kinematics are presented as follows.

Translational and rotational part of system dynamical equations are given in the standard form

$$m\dot{\mathbf{v}} - \mathbf{C}^T \boldsymbol{\lambda} = m\mathbf{g}, \tag{13}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} + \tilde{\mathbf{X}}\mathbf{R}^T \boldsymbol{\lambda} = 0, \tag{14}$$

where $\mathbf{x}$ is the body mass centre position, $\boldsymbol{\omega}$ represents body angular velocity, m and $\mathbf{J}$ are body mass and tensor of inertia, $\boldsymbol{\lambda}$ stands for joint reaction forces, $\mathbf{C}$ is constraint matrix, $\mathbf{g}$ is gravity, $\mathbf{X}$ is body mass centre in the local coordinate system fixed to the body and $\mathbf{R} \in SO(3)$ is rotation matrix that relates body coordinate system to inertial coordinate system.

Mechanical system constraints at position, velocity and acceleration level that represent body joint are given as

$$-\mathbf{x}+\mathbf{R}\mathbf{X}=0\,, \tag{15}$$

$$\begin{bmatrix}-\mathbf{I}_3 & -\mathbf{R}\tilde{\mathbf{X}}\end{bmatrix}\begin{bmatrix}\mathbf{v}\\ \boldsymbol{\omega}\end{bmatrix}=0\,, \tag{16}$$

$$\begin{bmatrix}-\mathbf{I}_3 & -\mathbf{R}\tilde{\mathbf{X}}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{v}}\\ \dot{\boldsymbol{\omega}}\end{bmatrix}=-\mathbf{R}\tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{X}\,, \tag{17}$$

and system constraint matrix can be shaped in the form

$$\mathbf{C}=\begin{bmatrix}-\mathbf{I}_3 & -\mathbf{R}\tilde{\mathbf{X}}\end{bmatrix}\;, \tag{18}$$

or

$$\mathbf{C}^T=\begin{bmatrix}-\mathbf{I}_3\\ -\tilde{\mathbf{X}}^T\mathbf{R}^T\end{bmatrix}=\begin{bmatrix}-\mathbf{I}_3\\ \tilde{\mathbf{X}}\mathbf{R}^T\end{bmatrix}, \tag{19}$$

which allows for assembling equations of system dynamics in DAE of index 1 form

$$\begin{bmatrix}m\mathbf{I}_3 & \mathbf{0} & -\mathbf{I}_3\\ \mathbf{0} & \mathbf{J} & \tilde{\mathbf{X}}\mathbf{R}^T\\ -\mathbf{I}_3 & -\mathbf{R}\tilde{\mathbf{X}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{v}}\\ \dot{\boldsymbol{\omega}}\\ \boldsymbol{\lambda}\end{bmatrix}=\begin{bmatrix}m\mathbf{g}\\ -\tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega}\\ -\mathbf{R}\tilde{\boldsymbol{\omega}}\tilde{\boldsymbol{\omega}}\mathbf{X}\end{bmatrix}. \tag{20}$$

The equation (20) has the same formal shape as (10) (as well as (A1) in [5], which has been formulated in vector space *via* generalized (local) coordinates). The numerical solution of (20) must satisfy (15), (16) and (17). The results of numerical integration are given by the Figures 1 - 4.

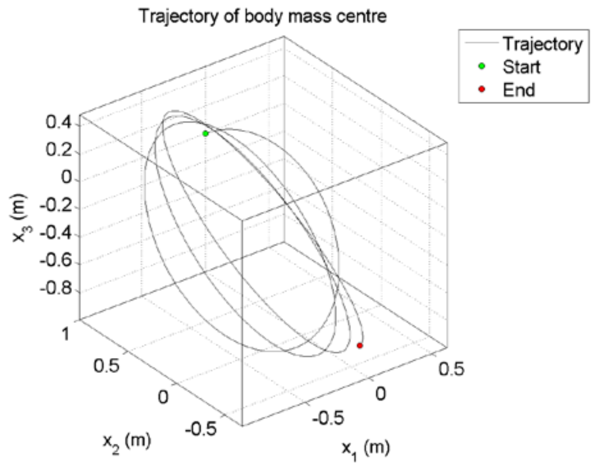

Figure 1: Spatial trajectory of body mass centre.

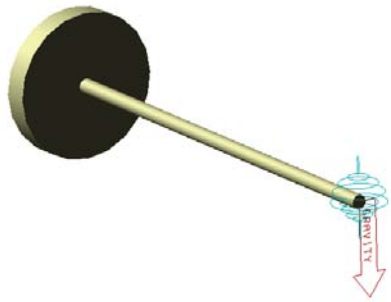

Figure 2: Sequence of the motion animation (post-processed *via* ADAMS).

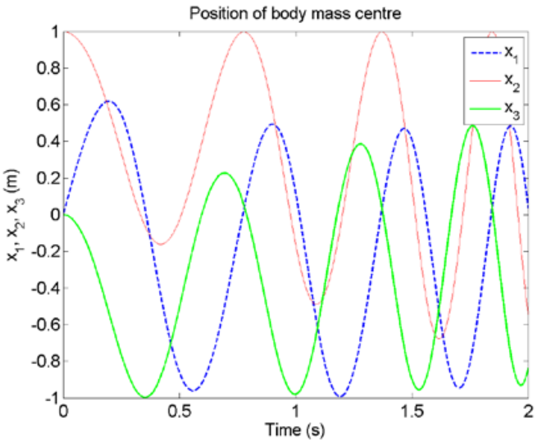

Figure 3: Position of body mass centre.

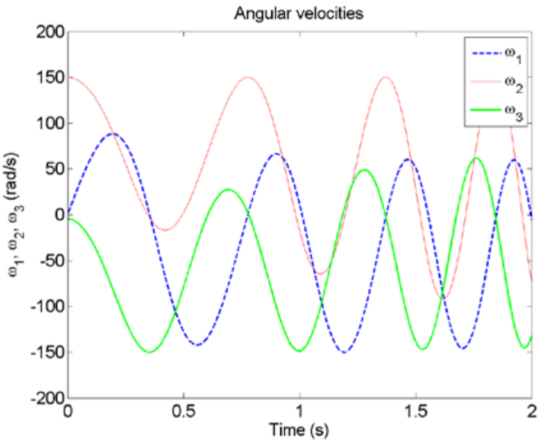

Figure 4: Body angular velocities.

3 CONCLUSION

The Lie-group integration method for constrained multibody systems is proposed in the paper. The method operates on Lie-group of system configuration that is modeled as 'state space formulation'. The system constraints are introduced in the mathematical model *via* DAE of index 1 fomulation and constraint violation is minimized by using constraint manifold projection method based on solving the nonlinear constrained least square problem. Since integration algorithm operates directly with angular velocities and rotational matrices, meaning that no local (generalized) coordinates are introduced, the method circumvent problems of kinematic singularities of rigid body three-parameters rotation basis, re-parameterization of system kinematics during integration as well as numerical non-efficiency of the kinematic differential equations. The method is numerically robust and it is easy-applicable on the general class of multibody systems.

REFERENCES

[1] Iserles, Munthe–Kaas, Norsett and Zanna. Lie-group methods. *Acta Numerica*, **9**, 215–365, 2000.

[2] E. Hairer, C. Lubich and G. Wanner. *Geometric Numerical Integration*. Springer, 2006.

[3] A. Müller. Approximation of finite rigid body motions from velocity fields. *Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, **90**, 514–521, 2010.

[4] O. Brüls and A. Cardona. On the Use of Lie Group Time Integrators in Multibody Dynamics. *Journal of Computational and Nonlinear Dynamic*, **5**, 2010.

[5] Z. Terze and J. Naudet. Geometric Properties of Projective Constraint Violation Stabilization Method for Generally Constrained Multibody Systems on Manifolds. *Multibody System Dynamics*, **20**, 85–106, 2008.

[6] E. Eich. Convergence Results for a Coordinate Projection Method Applied to Mechanical Systems with Algebraic Constraints, *SIAM J. Numer. Anal.*, **30**, 1467–1482, 1993.

[7] H. Munthe–Kaas. Runge–Kutta Methods on Lie Groups, *BIT*, **38**, 92–111, 1998.

[8] E. Celledoni, B. Owren. Lie Group Methods for Rigid Body Dynamics and Time Integration on Manifolds. *Comput. Methods Appl. Mech. Engrg.*, **192**, 421–438, 2003.

[9] E. Eich–Soellner, C. Führer. *Numerical Methods in Multibody Dynamics*. Teubner–Verlag, Stuttgart, 1998.

[10] Z. Terze and J. Naudet. Structure of Optimized Generalized Coordinates Partitioned Vectors for Holonomic and Non-holonomic Systems. *Multibody System Dynamics*, **24**, 203–218, 2010.

[11] A. Müller. Group Theoretical Approaches to Vector Parameterization of Rotations. *Journal of Geometry and Symmetry in Physics*, **4**, 1–30, 2005.

[12] A. Marthinsen, H. Munthe–Kaas and B. Owren. Simulation of Ordinary Differential Equations on Manifolds: Some Numerical Experiments and Verifications. *Modeling, Identification and Control*, **18**, 75–80, 1997.