

# A divide and conquer algorithm for constrained multibody system dynamics based on augmented Lagrangian method with projections-based error correction

Paweł Malczyk · Janusz Frączek

Received: 23 July 2011 / Accepted: 11 June 2012 / Published online: 3 July 2012  
© The Author(s) 2012. This article is published with open access at Springerlink.com

**Abstract** This paper presents a new parallel algorithm for dynamics simulation of general multibody systems. The developed formulations are iterative and possess divide and conquer structure. The constraints equations are imposed at the acceleration level. Augmented Lagrangian methods with mass-orthogonal projections are used to prevent from constraint violation errors. The proposed approaches treat tree topology mechanisms or multibody systems which contain kinematic closed loops in a uniform manner and can handle problems with rank deficient Jacobian matrices. Test case results indicate good accuracy performance dependent on the expense put in the iterative correction of constraint equations. Good numerical properties and robustness of the algorithms are observed when handling systems with single and coupled kinematic loops, redundant constraints, which may repeatedly enter singular configurations.

**Keywords** Divide and conquer algorithm · Augmented Lagrangian method · Mass-orthogonal projections · Parallel computing

## 1 Introduction

Efficient dynamics simulations of complex multibody systems (MBS) are of crucial importance in many areas of computer aided engineering and design. There are many examples of such systems including vehicles, biomechanical models, robots and multidisciplinary applications. Computations can be carried out by means of different types of formulations. To meet requirements for high-fidelity performance and accurate dynamics simulations of complex systems, it has become a practice to apply efficient, low order algorithms designed both for sequential and parallel computations.

The examples of efficient  $O(n)$  ( $n$ —number of bodies) recursive sequential algorithms for analysis of tree-like topology rigid body dynamics can be found in [1, 4, 5, 7, 14, 22, 36, 37, 41, 42]. Some researchers developed recursive order “ $O(n)$ ” formulations dynamics simulations of multibody systems with closed loops [2, 33, 38, 39]. These algorithms gave basis for further development of efficient low order formulations.

As parallel computing resources became more available, researchers began to adapt the existing formulations or design completely new algorithms, suitable for parallel computing. The strategies enabled to decrease the turnaround time associated with computer simulations and even achieve results in real-time. The first attempts to exploit parallel strategies can be found in [6, 12, 18, 19, 25, 26].

---

P. Malczyk (✉) · J. Frączek  
Institute of Aeronautics and Applied Mechanics,  
Faculty of Power and Aeronautical Engineering,  
Warsaw University of Technology, Nowowiejska 24,  
00-665 Warsaw, Poland  
e-mail: [pmalczyk@meil.pw.edu.pl](mailto:pmalczyk@meil.pw.edu.pl)

J. Frączek  
e-mail: [jfraczek@meil.pw.edu.pl](mailto:jfraczek@meil.pw.edu.pl)

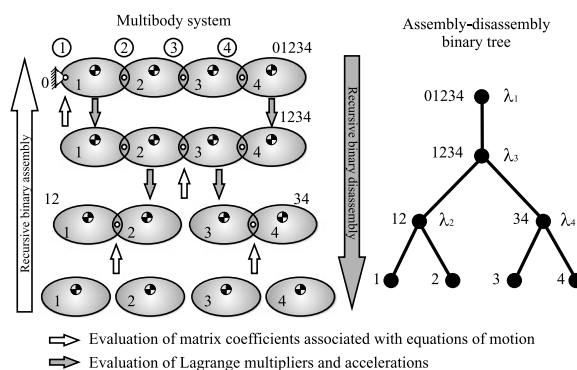
In the literature, there are more recent ideas regarding parallel algorithms for rigid multibody dynamics simulation. Featherstone [15, 16] developed the optimal-time, logarithmic order divide and conquer algorithm (DCA) for dynamics of general multibody systems. The idea behind the formulation lies in a recursive binary assembly and disassembly of articulated bodies. The extension to closed loop systems were obtained through the application of constraint stabilization methods and proper decomposition of constraint forces. Fisette and Peterkenne [20] and independently Anderson and Duan [3] adopted the idea of a decomposition of multibody system into subchains. In [13], the authors explored the ideas of recursive coordinate reduction and presented parallel multibody algorithm with optimal logarithmic time complexity for general multibody systems applicability. The formulation, however, was not free from problems arising from the fact that some matrices lost their ranks. Yamane and Nakamura [43] developed the new assembly-disassembly algorithm, which is based on the concept of divide and conquer scheme. In work [32], Mukherjee and Anderson presented exact and non-iterative divide and conquer algorithm for forward dynamics of MBS with single and coupled loops, which incorporated neither coordinate reductions nor Lagrange multipliers. The latest comparative study on efficient sequential and parallel multibody dynamics algorithms can be found in [44] and in recent books [17, 23].

Based on the divide and conquer scheme [15], the authors presented recently parallel algorithms for general constrained MBS dynamics using augmented Lagrangian methods and mass-orthogonal projections [27–31]. The improved, extended and verified versions of the methods are proposed in this paper. The developed parallel formulations for dynamics of general multibody systems encompass efficiency, accuracy and robustness in case of analyzing systems with redundant constraints containing single or multiple kinematic loops, and which may enter singular configurations.

## 2 Algorithm overview

### 2.1 Preliminaries

This section presents analytical derivations for the divide and conquer (DCA) algorithm based on aug-



**Fig. 1** Recursive binary assembly and disassembly of a four-link multibody system

mented Lagrangian formulation with projections for dynamics simulation of general multibody systems (MBS). The method used here is similar to that proposed by Featherstone [15, 16], however the underlying computations are different. The parallel algorithm is composed of only two computational stages: assembly and disassembly phase. Each computational phase is connected to the binary tree associated with the topology of the mechanism. The graphical representation of the example of assembly-disassembly process for a four-link MBS is shown in Fig. 1. The first stage starts with writing of equations of motion for each body in the system and is continued until the analyzed multibody system is assembled. Subassemblies, which correspond to nodes in the graph, are constructed by traversing the binary tree from leaves to the top node. Finally, a single assembly is obtained, which is denoted as a root node of the binary tree. This node constitutes a representation of the entire multibody system modeled as a single assembly. The main pass finishes at this stage. Taking into account the boundary conditions, e.g., a connection of a chain to fixed base and free floating terminal body, the second phase is started. Traversing the binary tree from the root node to leaves, all Lagrange multipliers and bodies' accelerations are computed in the system. This brief description of the DCA algorithm will help to understand the proposed formulations. The sequence of derivations will correspond to this short review.

## 2.2 Assembly phase

### 2.2.1 Equations of motion for two articulated bodies

To aid in the subsequent derivations, consider a system of two rigid bodies connected by a joint, as depicted in Fig. 2. Each body in the system is characterized by a  $7 \times 1$  vector of absolute coordinates  $\mathbf{q}_i = [\mathbf{r}_i^T \mathbf{p}_i^T]^T$ , where index  $i = 1, \dots, n$  is the number of bodies in the system. Vector  $\mathbf{r}_i$  describes a position of the origin (located in body's mass center) of the reference frame rigidly attached to body  $i$  with respect to global reference frame and  $\mathbf{p}_i$  is a vector of Euler parameters representing orientation of the local frame. Moreover, Lagrange multipliers  $\lambda_1, \lambda_2, \lambda$  are associated with constraint forces in joints (Fig. 2).

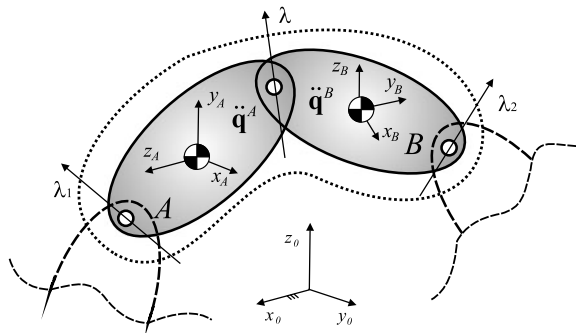
Equations of motion for bodies  $A$  and  $B$  can be written in the form, which takes into account Euler parameter representation of rotations:

$$\mathbf{M}^A \ddot{\mathbf{q}}^A + \Phi_{\mathbf{q}^A}^T \lambda_1 + \Phi_{\mathbf{q}^A}^T \lambda + \Phi_{\mathbf{q}^A}^N \lambda_A^N = \mathbf{Q}^A, \quad (1)$$

$$\mathbf{M}^B \ddot{\mathbf{q}}^B + \Phi_{\mathbf{q}^B}^T \lambda + \Phi_{\mathbf{q}^B}^T \lambda_2 + \Phi_{\mathbf{q}^B}^N \lambda_B^N = \mathbf{Q}^B. \quad (2)$$

where the inertia matrices  $\mathbf{M}^A$  and  $\mathbf{M}^B$  are  $7 \times 7$  quantities, which are known to be singular [21, 34], vectors  $\mathbf{Q}^A, \mathbf{Q}^B$  are applied, position and velocity dependent forces, Lagrange multipliers  $\lambda_1, \lambda_2, \lambda$  are associated with joint constraint equations  $\Phi^1, \Phi^2, \Phi$ , and  $\lambda_A^N, \lambda_B^N$  are Lagrange multipliers related to Euler parameters normalization constraints  $\Phi^N$  for body  $A$  and  $B$ . The joint between the bodies imposes the constraint equations that could be expressed at the acceleration level as follows:

$$\ddot{\Phi} = \Phi_{\mathbf{q}^A} \ddot{\mathbf{q}}^A + \Phi_{\mathbf{q}^B} \ddot{\mathbf{q}}^B + \gamma^{AB} = 0, \quad (3)$$



**Fig. 2** Two articulated bodies

where vector  $\gamma^{AB}$  is that part of acceleration constraint equations which are dependent on positions and velocities. The Euler parameters normalization constraints can be also written as a second derivatives with respect to time:

$$\ddot{\Phi}_A^N = \Phi_{\mathbf{q}^A}^N \ddot{\mathbf{q}}^A + \gamma_A^N = 0, \quad (4)$$

$$\ddot{\Phi}_B^N = \Phi_{\mathbf{q}^B}^N \ddot{\mathbf{q}}^B + \gamma_B^N = 0. \quad (5)$$

To apply the divide and conquer scheme, it is required to evaluate the accelerations of bodies  $A$  and  $B$  (see [15, 29]). The matrices  $\mathbf{M}^A$  and  $\mathbf{M}^B$  are not invertible in Eqs. (1) and (2). To overcome this difficulty, augmented Lagrangian formulation is employed [10, 11, 24, 35], which leads to the following Lagrange multipliers iterative approximation:

$$\lambda = \lambda^* + \alpha (\Phi_{\mathbf{q}^A} \ddot{\mathbf{q}}^A + \Phi_{\mathbf{q}^B} \ddot{\mathbf{q}}^B + \underbrace{\gamma^{AB}}_{\bar{\gamma}^{AB}}), \quad (6)$$

$$\lambda_A^N = \lambda_A^{N*} + \alpha (\Phi_{\mathbf{q}^A}^N \ddot{\mathbf{q}}^A + \underbrace{\gamma_A^N + 2\Omega \mu \dot{\Phi}_A^N + \Omega^2 \Phi_A^N}_{\bar{\gamma}_A^N}), \quad (7)$$

$$\lambda_B^N = \lambda_B^{N*} + \alpha (\Phi_{\mathbf{q}^B}^N \ddot{\mathbf{q}}^B + \underbrace{\gamma_B^N + 2\Omega \mu \dot{\Phi}_B^N + \Omega^2 \Phi_B^N}_{\bar{\gamma}_B^N}), \quad (8)$$

where  $\lambda^*, \lambda_A^{N*}$  and  $\lambda_B^{N*}$  are Lagrange multipliers obtained from the previous iteration and multipliers without star sign are related to the current iteration. We assume that  $\lambda^* = 0, \lambda_A^{N*} = 0, \lambda_B^{N*} = 0$  for the initial iteration. The matrices  $\alpha = \alpha \mathbf{I} = \text{diag}(\alpha)$ ,  $\mu$  and  $\Omega$  are diagonal matrices that contain values of the large penalty numbers, natural frequencies and damping ratios of the system added to each constraint to fulfill constraint equations [9–11]. Let us substitute Eqs. (6), (7), and (8) to Eqs. (1) and (2) to obtain the following result:

$$\begin{aligned} & (\mathbf{M}^A + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^A} + \Phi_{\mathbf{q}^A}^N \alpha \Phi_{\mathbf{q}^A}^N) \ddot{\mathbf{q}}^A \\ & + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^B} \ddot{\mathbf{q}}^B + \Phi_{\mathbf{q}^A}^T \lambda_1 \\ & = \mathbf{Q}^A - \Phi_{\mathbf{q}^A}^T (\lambda^* + \alpha \bar{\gamma}^{AB}) \\ & - \Phi_{\mathbf{q}^A}^N (\lambda_A^{N*} + \alpha \bar{\gamma}_A^N), \end{aligned} \quad (9)$$

$$\begin{aligned}
& \Phi_{q^B}^T \alpha \Phi_{q^A} \ddot{q}^A + (M^B + \Phi_{q^B}^T \alpha \Phi_{q^B} \\
& \quad + \Phi_{q^B}^N{}^T \alpha \Phi_{q^B}^N) \ddot{q}^B + \Phi_{q^B}^2{}^T \lambda_2 \\
& = Q^B - \Phi_{q^B}^T (\lambda^* + \alpha \bar{\gamma}^{AB}) \\
& \quad - \Phi_{q^B}^N{}^T (\lambda_B^{N*} + \alpha \bar{\gamma}_B^N). \quad (10)
\end{aligned}$$

Equations (9) and (10) form a basis for further derivations. They are starting point for the main pass phase. In contrast to Eqs. (1) and (2), relations (9) and (10) can be solved for  $\ddot{q}^A$  and  $\ddot{q}^B$ , respectively. The encountered leading matrices are symmetric and positive definite, therefore, they are invertible, even in situations when constraint Jacobian matrices lose their rank.

### 2.2.2 Equations of motion for the system of articulated bodies

The procedure of elimination of Lagrange multipliers between two bodies presented in the preceding subsection can be generalized and repeated for the larger set of articulated bodies. Consider a system of rigid bodies  $A$  and  $B$  interconnected by joints, as shown in Fig. 3.

The system is under influence of various acceleration-independent forces and its state is known. Three Lagrange multipliers associated with joints are specified. Those with subscript 1 and 2 are used to interact with other subsystems in the multibody system (MBS) and the multipliers between the articulated bodies  $A$  and  $B$  serve as those, which are eliminated during the computations. The generalization of Eqs. (9) and (10) may be written as follows:

$$M_{11}^A \ddot{q}_1^A + M_{12}^A \ddot{q}_2^A + \Phi_{q_1^A}^1{}^T \lambda_1 = Q_1^A, \quad (11)$$

$$M_{21}^A \ddot{q}_1^A + M_{22}^A \ddot{q}_2^A + \Phi_{q_2^A}^T \lambda = Q_2^A, \quad (12)$$

and

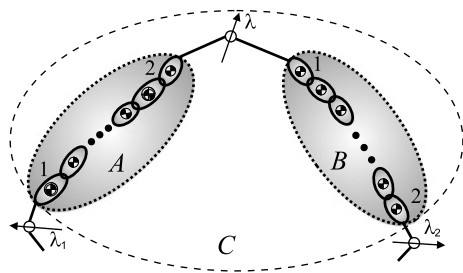


Fig. 3 System of articulated rigid bodies

$$M_{11}^B \ddot{q}_1^B + M_{12}^B \ddot{q}_2^B + \Phi_{q_1^B}^T \lambda = Q_1^B, \quad (13)$$

$$M_{21}^B \ddot{q}_1^B + M_{22}^B \ddot{q}_2^B + \Phi_{q_2^B}^2{}^T \lambda_2 = Q_2^B. \quad (14)$$

The objective of the following algebraic manipulations is to obtain equations of motion for the set  $C$ , in the form:

$$M_{11}^C \ddot{q}_1^C + M_{12}^C \ddot{q}_2^C + \Phi_{q_1^C}^1{}^T \lambda_1 = Q_1^C, \quad (15)$$

$$M_{21}^C \ddot{q}_1^C + M_{22}^C \ddot{q}_2^C + \Phi_{q_2^C}^2{}^T \lambda_2 = Q_2^C. \quad (16)$$

Again, unknown Lagrange multipliers between the sets of bodies  $A$  and  $B$  can be found from the iterative process:

$$\lambda = \lambda^* + \alpha (\Phi_{q_2^A} \ddot{q}_2^A + \Phi_{q_1^B} \ddot{q}_1^B + \bar{\gamma}^{AB}). \quad (17)$$

Multiplying Eq. (17) by  $\alpha^{-1} = \frac{1}{\alpha} \mathbf{I}$  and inserting accelerations from Eqs. (12), (13) to Eq. (17) yields:

$$\begin{aligned}
\frac{1}{\alpha} \mathbf{I} \lambda &= \frac{1}{\alpha} \mathbf{I} \lambda^* + \Phi_{q_2^A} (M_{22}^A)^{-1} (Q_2^A - M_{21}^A \ddot{q}_1^A - \Phi_{q_2^A}^T \lambda) \\
& \quad + \Phi_{q_1^B} (M_{11}^B)^{-1} (Q_1^B - M_{12}^B \ddot{q}_2^B - \Phi_{q_1^B}^T \lambda) \\
& \quad + \bar{\gamma}^{AB}, \quad (18)
\end{aligned}$$

where  $\mathbf{I}$  is the identity matrix of the proper dimension. The relations (18) can be solved in terms of Lagrange multipliers between the sets of bodies  $A$  and  $B$ :

$$\begin{aligned}
& \left( \frac{1}{\alpha} \mathbf{I} + \Phi_{q_2^A} (M_{22}^A)^{-1} \Phi_{q_2^A}^T + \Phi_{q_1^B} (M_{11}^B)^{-1} \Phi_{q_1^B}^T \right) \lambda \\
& = \frac{1}{\alpha} \mathbf{I} \lambda^* + \Phi_{q_2^A} (M_{22}^A)^{-1} (Q_2^A - M_{21}^A \ddot{q}_1^A) \\
& \quad + \Phi_{q_1^B} (M_{11}^B)^{-1} (Q_1^B - M_{12}^B \ddot{q}_2^B) + \bar{\gamma}^{AB}. \quad (19)
\end{aligned}$$

Let us define a matrix  $C$  as follows:

$$\begin{aligned}
C &= \left( \frac{1}{\alpha} \mathbf{I} + \Phi_{q_2^A} (M_{22}^A)^{-1} \Phi_{q_2^A}^T \right. \\
& \quad \left. + \Phi_{q_1^B} (M_{11}^B)^{-1} \Phi_{q_1^B}^T \right)^{-1}. \quad (20)
\end{aligned}$$

Closer look at matrix in Eq. (20) reveals that the inversion always exists, even when constraint Jacobian matrices become rank deficient. Substituting Eq. (19) into Eq. (12) and taking into account Eq. (20), we obtain:

$$\begin{aligned}
\ddot{q}_2^A &= (M_{22}^A)^{-1} \{ \bar{Q}_2^A \\
& \quad - (\mathbf{I} - \Phi_{q_2^A}^T C \Phi_{q_2^A} (M_{22}^A)^{-1}) M_{21}^A \ddot{q}_1^A \\
& \quad + \Phi_{q_2^A}^T C \Phi_{q_1^B} (M_{11}^B)^{-1} M_{12}^B \ddot{q}_2^B \}, \quad (21)
\end{aligned}$$

where

$$\begin{aligned} \overline{\mathbf{Q}}_2^A = \mathbf{Q}_2^A - \Phi_{q_2^A}^T \mathbf{C} \left( \frac{1}{\alpha} \mathbf{I}^* + \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1} \mathbf{Q}_2^A \right. \\ \left. + \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1} \mathbf{Q}_1^B + \bar{\mathbf{y}}^{AB} \right). \end{aligned} \quad (22)$$

Similarly, we can insert Eq. (19) into Eq. (13) taking into account Eq. (20):

$$\begin{aligned} \ddot{\mathbf{q}}_1^B = (\mathbf{M}_{11}^B)^{-1} \{ \overline{\mathbf{Q}}_1^B \\ - (\mathbf{I} - \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1}) \mathbf{M}_{12}^B \ddot{\mathbf{q}}_2^B \\ + \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1} \mathbf{M}_{21}^A \ddot{\mathbf{q}}_1^A \}, \end{aligned} \quad (23)$$

where

$$\begin{aligned} \overline{\mathbf{Q}}_1^B = \mathbf{Q}_1^B - \Phi_{q_1^B}^T \mathbf{C} \left( \frac{1}{\alpha} \mathbf{I}^* + \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1} \mathbf{Q}_2^A \right. \\ \left. + \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1} \mathbf{Q}_1^B + \bar{\mathbf{y}}^{AB} \right). \end{aligned} \quad (24)$$

The final step is to substitute accelerations from Eqs. (21) and (23) into Eqs. (11) and (14), respectively:

$$\begin{aligned} (\mathbf{M}_{11}^A - \mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} (\mathbf{I} - \Phi_{q_2^A}^T \mathbf{C} \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1}) \mathbf{M}_{21}^A) \ddot{\mathbf{q}}_1^A \\ + (\mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} \Phi_{q_2^A}^T \mathbf{C} \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1} \mathbf{M}_{12}^B) \ddot{\mathbf{q}}_2^B \\ + \Phi_{q_1^A}^T \lambda_1 = \mathbf{Q}_1^A - \mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} \overline{\mathbf{Q}}_2^A, \end{aligned} \quad (25)$$

$$\begin{aligned} (\mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1} \mathbf{M}_{21}^A) \ddot{\mathbf{q}}_1^A + \Phi_{q_2^B}^T \lambda_2 \\ + (\mathbf{M}_{22}^B - \mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \\ \times (\mathbf{I} - \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1}) \mathbf{M}_{12}^B) \ddot{\mathbf{q}}_2^B \\ = \mathbf{Q}_2^B - \mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \overline{\mathbf{Q}}_1^B. \end{aligned} \quad (26)$$

By gathering appropriate matrix coefficients and comparing Eq. (15) and (16) together with Eqs. (25) and (26), the following relations are obtained:

$$\begin{aligned} \mathbf{M}_{11}^C = \mathbf{M}_{11}^A - \mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} \\ \times (\mathbf{I} - \Phi_{q_2^A}^T \mathbf{C} \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1}) \mathbf{M}_{21}^A, \end{aligned} \quad (27)$$

$$\mathbf{M}_{12}^C = \mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} \Phi_{q_2^A}^T \mathbf{C} \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1} \mathbf{M}_{12}^B, \quad (28)$$

$$\begin{aligned} \mathbf{M}_{21}^C = \mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_2^A} (\mathbf{M}_{22}^A)^{-1} \mathbf{M}_{21}^A \\ = (\mathbf{M}_{12}^C)^T, \end{aligned} \quad (29)$$

$$\begin{aligned} \mathbf{M}_{22}^C = \mathbf{M}_{22}^B - \mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \\ \times (\mathbf{I} - \Phi_{q_1^B}^T \mathbf{C} \Phi_{q_1^B} (\mathbf{M}_{11}^B)^{-1}) \mathbf{M}_{12}^B, \end{aligned} \quad (30)$$

and

$$\mathbf{Q}_1^C = \mathbf{Q}_1^A - \mathbf{M}_{12}^A (\mathbf{M}_{22}^A)^{-1} \overline{\mathbf{Q}}_2^A, \quad (31)$$

$$\mathbf{Q}_2^C = \mathbf{Q}_2^B - \mathbf{M}_{21}^B (\mathbf{M}_{11}^B)^{-1} \overline{\mathbf{Q}}_1^B. \quad (32)$$

Relations (27) to (32) are used for the first computational stage called assembly phase. The recursive formulas are exploited until the whole MBS is constructed according to the binary tree associated with mechanism decomposition.

### 2.3 Disassembly phase

The assembly phase finishes when the root node of the binary tree is achieved. At this computational stage, the equations of motion for the whole multibody system can be expressed in the general form, which is similar to Eq. (15) and Eq. (16):

$$\mathbf{M}_{11}^C \ddot{\mathbf{q}}_1^C + \mathbf{M}_{12}^C \ddot{\mathbf{q}}_2^C + \Phi_{q_1^C}^T \lambda_1 = \mathbf{Q}_1^C, \quad (33)$$

$$\mathbf{M}_{21}^C \ddot{\mathbf{q}}_1^C + \mathbf{M}_{22}^C \ddot{\mathbf{q}}_2^C + \Phi_{q_2^C}^T \lambda_2 = \mathbf{Q}_2^C. \quad (34)$$

The quantity  $\ddot{\mathbf{q}}_1^C$  is the acceleration of the first body in the system (body 1 from the set  $A$  in Fig. 3) and  $\ddot{\mathbf{q}}_2^C$  is the acceleration of the last body (body 2 from the set  $B$  in Fig. 3). Lagrange multipliers  $\lambda_1$  and  $\lambda_2$  correspond to constraints connecting first and last body with topological parent and child bodies, respectively, provided they exist. There are three different boundary cases associated with the root node of the binary tree:

1.  $\lambda_1 = \mathbf{0}, \lambda_2 = \mathbf{0}$ .

Taking into account Eqs. (33) and (34), the accelerations can be evaluated from the following system of linear equations:

$$\begin{bmatrix} \mathbf{M}_{11}^C & \mathbf{M}_{12}^C \\ \mathbf{M}_{21}^C & \mathbf{M}_{22}^C \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1^C \\ \ddot{\mathbf{q}}_2^C \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1^C \\ \mathbf{Q}_2^C \end{bmatrix}. \quad (35)$$

The matrix coefficients  $\mathbf{M}_{11}^C$  and  $\mathbf{M}_{22}^C$  are symmetric and positive definite, moreover  $\mathbf{M}_{12}^C = (\mathbf{M}_{21}^C)^T$ . The accelerations  $\ddot{\mathbf{q}}_1^C$  and  $\ddot{\mathbf{q}}_2^C$  can be easily evaluated from Eq. (35) to begin disassembly phase. The case can be used for the simulation of free-floating systems.

2.  $\lambda_1 \neq \mathbf{0}$  i  $\lambda_2 = \mathbf{0}$  or inversely.

The Lagrange multipliers  $\lambda_1$  can be approximated from the process (6) or (17) as follows:

$$\lambda_1 = \lambda_1^* + \alpha(\Phi_{q_1^C}^1 \ddot{q}_1^C + \bar{\gamma}_1^{AB}). \quad (36)$$

Inserting Eq. (36) to (33) and taking into account  $\lambda_2 = \mathbf{0}$  yields:

$$\begin{bmatrix} \mathbf{M}_{11}^C + \Phi_{q_1^C}^1{}^T \alpha \Phi_{q_1^C}^1 & \mathbf{M}_{12}^C \\ \mathbf{M}_{21}^C & \mathbf{M}_{22}^C \end{bmatrix} \begin{bmatrix} \ddot{q}_1^C \\ \ddot{q}_2^C \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1^C - \Phi_{q_1^C}^1{}^T (\lambda_1^* + \alpha \bar{\gamma}_1^{AB}) \\ \mathbf{Q}_2^C \end{bmatrix}. \quad (37)$$

Again the leading matrix of coefficients from the linear system of Eqs. (37) is invertible. The case can be applied to the simulation of various tree-like multibody systems.

3.  $\lambda_1 \neq \mathbf{0}$  and  $\lambda_2 \neq \mathbf{0}$ .

The Lagrange multipliers  $\lambda_1$  and  $\lambda_2$  can be obtained from the approximation similar to that in Eq. (36):

$$\lambda_1 = \lambda_1^* + \alpha(\Phi_{q_1^C}^1 \ddot{q}_1^C + \bar{\gamma}_1^{AB}), \quad (38)$$

$$\lambda_2 = \lambda_2^* + \alpha(\Phi_{q_2^C}^2 \ddot{q}_2^C + \bar{\gamma}_2^{AB}). \quad (39)$$

Inserting Eq. (38) into (33) and Eq. (39) into (34) yields:

$$\begin{bmatrix} \mathbf{M}_{11}^C + \Phi_{q_1^C}^1{}^T \alpha \Phi_{q_1^C}^1 & \mathbf{M}_{12}^C \\ \mathbf{M}_{21}^C & \mathbf{M}_{22}^C + \Phi_{q_2^C}^2{}^T \alpha \Phi_{q_2^C}^2 \end{bmatrix} \times \begin{bmatrix} \ddot{q}_1^C \\ \ddot{q}_2^C \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1^C - \Phi_{q_1^C}^1{}^T (\lambda_1^* + \alpha \bar{\gamma}_1^{AB}) \\ \mathbf{Q}_2^C - \Phi_{q_2^C}^2{}^T (\lambda_2^* + \alpha \bar{\gamma}_2^{AB}) \end{bmatrix}. \quad (40)$$

Again the matrix coefficients in linear system of Eqs. (40) are symmetric and positive definite, thus they are invertible. This case can be applied for analysis of general closed loop multibody systems.

The process of assembly starts with equations of motion for individual bodies in the system and is continued until the whole mechanism is composed. Sub-assemblies, which correspond to nodes in the graph, are constructed by traversing the binary tree, according to Eqs. (27)–(32). Finally, the procedure for solving the equations of motion of the root node using

the boundary conditions described in this subsection is employed. Traversing the tree from root node to leaves, all accelerations are computed according to Eq. (21) and Eq. (23). Moreover, if it is needed, Lagrange multipliers can be evaluated using Eq. (17) or Eq. (19).

### 3 Mass-orthogonal projections

#### 3.1 Introduction

Derived parallel divide and conquer scheme possesses good overall characteristics and robustness in case of general multibody dynamics simulations. However, the formulation does not enforce simultaneously the error control on position and velocity constraints. In order to overcome these drawbacks, mass-orthogonal projections in positions and velocities are employed for further improvements [10, 11]. The process of correction of the state variables is numerically expensive. The projections in positions and velocities are expressed in a form of separate algorithms, which maintain the divide and conquer structure [27]. The Appendix accompanying the paper may be helpful in understanding some of the following relations.

#### 3.2 Projections in positions

During the integration process, vector of position coordinates  $\mathbf{q}_*$  may not satisfy position constraint equations. In order to correct bodies' positions, mass-orthogonal projections of the solution to the constraint manifold are performed [10]. Consider again the system of two rigid bodies connected by a joint, as depicted in Fig. 2. The projections in positions [10, 27] can be written in a form, which is similar to that expressed in Eqs. (1) and (2) (see Eq. (93) in the Appendix):

$$\mathbf{M}^A \Delta \mathbf{q}^A + \Phi_{q^A}^1{}^T \lambda_1 + \Phi_{q^A}^T \lambda + \Phi_{q^A}^N{}^T \lambda_A^N = -\mathbf{M}^A (\mathbf{q}^A - \mathbf{q}_*^A), \quad (41)$$

$$\mathbf{M}^B \Delta \mathbf{q}^B + \Phi_{q^B}^T \lambda + \Phi_{q^B}^2{}^T \lambda_2 + \Phi_{q^B}^N{}^T \lambda_B^N = -\mathbf{M}^B (\mathbf{q}^B - \mathbf{q}_*^B), \quad (42)$$

where vectors  $\lambda$  (with and without subscripts or superscripts) are associated with the problem of constrained minimization of the functional

$$V = \frac{1}{2} (\mathbf{q} - \mathbf{q}_*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}_*)$$



subjected to position constraint equations  $\Phi(\mathbf{q}, t) = \mathbf{0}$  [10, 27], whereas  $\Delta \mathbf{q}^A$  and  $\Delta \mathbf{q}^B$  are unknown position corrections. Constraint equations between body  $A$  and  $B$  can be written as

$$\Phi(\mathbf{q}^A, \mathbf{q}^B, t) = \mathbf{0}. \quad (43)$$

Euler parameters normalization constraint equations for body  $A$  and  $B$  can be expressed as

$$\Phi_A^N(\mathbf{q}^A) = 0, \quad \Phi_B^N(\mathbf{q}^B) = 0. \quad (44)$$

Lagrange multipliers in Eqs. (41) and (42) can be evaluated by using the following iterative approximation (see Eq. (100) in the Appendix):

$$\begin{aligned} \lambda &= \lambda^* + \alpha \Phi(\mathbf{q}^A, \mathbf{q}^B, t) \\ &\approx \lambda^* + \alpha(\Phi_{\mathbf{q}^A} \Delta \mathbf{q}^A + \Phi_{\mathbf{q}^B} \Delta \mathbf{q}^B \\ &\quad + \Phi^*(\mathbf{q}_*^A, \mathbf{q}_*^B)), \end{aligned} \quad (45)$$

$$\begin{aligned} \lambda_A^N &= \lambda_A^{N*} + \alpha \Phi_A^N(\mathbf{q}^A) \\ &\approx \lambda_A^{N*} + \alpha(\Phi_{\mathbf{q}^A}^N \Delta \mathbf{q}^A + \Phi_A^N(\mathbf{q}_*^A)), \end{aligned} \quad (46)$$

$$\begin{aligned} \lambda_B^N &= \lambda_B^{N*} + \alpha \Phi_B^N(\mathbf{q}^B) \\ &\approx \lambda_B^{N*} + \alpha(\Phi_{\mathbf{q}^B}^N \Delta \mathbf{q}^B + \Phi_B^N(\mathbf{q}_*^B)), \end{aligned} \quad (47)$$

where vectors  $\lambda$  with the asterisk sign are quantities evaluated in previous iteration. We assume  $\lambda^* = \mathbf{0}$ ,  $\lambda_A^{N*} = 0$ ,  $\lambda_B^{N*} = 0$  for initial iteration. Inserting Eqs. (45), (46), and (47) into Eqs. (41) and (42) yields:

$$\begin{aligned} &(\mathbf{M}^A + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^A} + \Phi_{\mathbf{q}^A}^N \alpha \Phi_{\mathbf{q}^A}^N) \Delta \mathbf{q}^A \\ &\quad + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^B} \Delta \mathbf{q}^B + \Phi_{\mathbf{q}^A}^1 \lambda_1 \\ &= -\mathbf{M}^A(\mathbf{q}^A - \mathbf{q}_*^A) - \Phi_{\mathbf{q}^A}^T (\lambda^* + \alpha \Phi^*(\mathbf{q}_*^A, \mathbf{q}_*^B)) \\ &\quad - \Phi_{\mathbf{q}^A}^N (\lambda_A^{N*} + \alpha \Phi_A^N(\mathbf{q}_*^A)), \end{aligned} \quad (48)$$

$$\begin{aligned} &\Phi_{\mathbf{q}^B}^T \alpha \Phi_{\mathbf{q}^A} \Delta \mathbf{q}^A \\ &\quad + (\mathbf{M}^B + \Phi_{\mathbf{q}^B}^T \alpha \Phi_{\mathbf{q}^B} + \Phi_{\mathbf{q}^B}^N \alpha \Phi_{\mathbf{q}^B}^N) \Delta \mathbf{q}^B \\ &\quad + \Phi_{\mathbf{q}^B}^2 \lambda_2 \\ &= -\mathbf{M}^B(\mathbf{q}^B - \mathbf{q}_*^B) - \Phi_{\mathbf{q}^B}^T (\lambda^* + \alpha \Phi^*(\mathbf{q}_*^A, \mathbf{q}_*^B)) \\ &\quad - \Phi_{\mathbf{q}^B}^N (\lambda_B^{N*} + \alpha \Phi_B^N(\mathbf{q}_*^B)). \end{aligned} \quad (49)$$

It should be emphasized that matrix coefficients at position increments are similar to that expressed in Eqs. (9), (10) and possess the same features. During the projections in positions, these values have to be evaluated only once per integration step. Equations

(48) and (49) can be generalized for the system of articulated bodies from Fig. 3 as follows:

$$\mathbf{M}_{11}^A \Delta \mathbf{q}_1^A + \mathbf{M}_{12}^A \Delta \mathbf{q}_2^A + \Phi_{\mathbf{q}_1^A}^1 \lambda_1 = \mathbf{Q}_1^A, \quad (50)$$

$$\mathbf{M}_{21}^A \Delta \mathbf{q}_1^A + \mathbf{M}_{22}^A \Delta \mathbf{q}_2^A + \Phi_{\mathbf{q}_2^A}^T \lambda = \mathbf{Q}_2^A, \quad (51)$$

and

$$\mathbf{M}_{11}^B \Delta \mathbf{q}_1^B + \mathbf{M}_{12}^B \Delta \mathbf{q}_2^B + \Phi_{\mathbf{q}_1^B}^T \lambda = \mathbf{Q}_1^B, \quad (52)$$

$$\mathbf{M}_{21}^B \Delta \mathbf{q}_1^B + \mathbf{M}_{22}^B \Delta \mathbf{q}_2^B + \Phi_{\mathbf{q}_2^B}^2 \lambda_2 = \mathbf{Q}_2^B. \quad (53)$$

The objective of the algorithm at the position level is to find relations, which describe articulated body  $C$  (Fig. 3) in the form:

$$\mathbf{M}_{11}^C \Delta \mathbf{q}_1^C + \mathbf{M}_{12}^C \Delta \mathbf{q}_2^C + \Phi_{\mathbf{q}_1^C}^1 \lambda_1 = \mathbf{Q}_1^C, \quad (54)$$

$$\mathbf{M}_{21}^C \Delta \mathbf{q}_1^C + \mathbf{M}_{22}^C \Delta \mathbf{q}_2^C + \Phi_{\mathbf{q}_2^C}^2 \lambda_2 = \mathbf{Q}_2^C. \quad (55)$$

Lagrange multipliers between sets of bodies  $A$  and  $B$  can be evaluated using the following iterative scheme (see Eq. (100) in the Appendix):

$$\lambda = \lambda^* + \alpha(\Phi_{\mathbf{q}_2^A} \Delta \mathbf{q}_2^A + \Phi_{\mathbf{q}_1^B} \Delta \mathbf{q}_1^B + \Phi^*(\mathbf{q}_*^A, \mathbf{q}_*^B)). \quad (56)$$

Relations (50)–(56) take on the same form of Eqs. (11)–(17). Consistently, the assembly phase can be performed by using relations (27)–(32). The matrix coefficients (27)–(30) are computed once per integration step and are exploited in the velocity projections as well as in the process of acceleration evaluation. On the other hand vector quantities  $\mathbf{Q}_i^A$ ,  $\mathbf{Q}_i^B$  for  $i = 1, 2$  in Eqs. (31)–(32) are computed in each iteration. The iterative scheme is continued until the stop criterion is met, which can be formulated as  $\|\Delta \mathbf{q}\| < \epsilon_p$ , where  $\epsilon_p$  is the user specified tolerance.

### 3.3 Projections in velocities

Similarly as at the position level, the numerical integration procedure yields velocities  $\dot{\mathbf{q}}_*$ , which may not satisfy velocity constraint equations with the desired accuracy. The mass-orthogonal velocity projections to the constraint manifold help to improve numerical accuracy at this level. Let us consider the system of two rigid bodies, depicted in Fig. 2. The projections in velocities can be written in the following form [10, 27] (see Eq. (104) in the Appendix):

$$\begin{aligned} \mathbf{M}^A \dot{\mathbf{q}}^A + \Phi_{\mathbf{q}^A}^1 T \lambda_1 + \Phi_{\mathbf{q}^A}^T \lambda + \Phi_{\mathbf{q}^A}^N T \lambda_A^N \\ = \mathbf{M}^A \dot{\mathbf{q}}_*^A, \end{aligned} \quad (57)$$

$$\begin{aligned} \mathbf{M}^B \dot{\mathbf{q}}^B + \Phi_{\mathbf{q}^B}^T \lambda + \Phi_{\mathbf{q}^B}^2 T \lambda_2 + \Phi_{\mathbf{q}^B}^N T \lambda_B^N \\ = \mathbf{M}^B \dot{\mathbf{q}}_*^B, \end{aligned} \quad (58)$$

where vectors  $\lambda$  (with and without subscripts or superscripts) correspond to the process of constrained minimization of the functional  $V = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_*)^T \mathbf{M}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_*)$  subjected to velocity constraint equations  $\dot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$ . The constraint equations between body  $A$  and  $B$  and Euler parameter normalization constraints at the velocity level can be written as follows:

$$\begin{aligned} \dot{\Phi}(\mathbf{q}^A, \mathbf{q}^B, \dot{\mathbf{q}}^A, \dot{\mathbf{q}}^B, t) \\ = \Phi_{\mathbf{q}^A} \dot{\mathbf{q}}^A + \Phi_{\mathbf{q}^B} \dot{\mathbf{q}}^B + \Phi_t^{AB} = \mathbf{0}, \end{aligned} \quad (59)$$

$$\dot{\Phi}_A^N(\mathbf{q}^A, \dot{\mathbf{q}}^A) = \Phi_{\mathbf{q}^A}^N \dot{\mathbf{q}}^A = \mathbf{0}, \quad (60)$$

$$\dot{\Phi}_B^N(\mathbf{q}^B, \dot{\mathbf{q}}^B) = \Phi_{\mathbf{q}^B}^N \dot{\mathbf{q}}^B = \mathbf{0}.$$

The Lagrange multipliers at the velocity level can be approximated as (see Eq. (110) in the Appendix):

$$\begin{aligned} \lambda = \lambda^* + \alpha \dot{\Phi}(\mathbf{q}^A, \mathbf{q}^B, \dot{\mathbf{q}}^A, \dot{\mathbf{q}}^B, t) \\ = \lambda^* + \alpha(\Phi_{\mathbf{q}^A} \dot{\mathbf{q}}^A + \Phi_{\mathbf{q}^B} \dot{\mathbf{q}}^B + \Phi_t^{AB}), \end{aligned} \quad (61)$$

$$\lambda_A^N = \lambda_A^{N*} + \alpha \dot{\Phi}_A^N(\mathbf{q}^A, \dot{\mathbf{q}}^A) = \lambda_A^{N*} + \alpha \Phi_{\mathbf{q}^A}^N \dot{\mathbf{q}}^A, \quad (62)$$

$$\lambda_B^N = \lambda_B^{N*} + \alpha \dot{\Phi}_B^N(\mathbf{q}^B, \dot{\mathbf{q}}^B) = \lambda_B^{N*} + \alpha \Phi_{\mathbf{q}^B}^N \dot{\mathbf{q}}^B, \quad (63)$$

where Lagrange multipliers  $\lambda$  with the asterisk sign are evaluated in previous iteration. We assume  $\lambda^* = \mathbf{0}$ ,  $\lambda_A^{N*} = \mathbf{0}$ ,  $\lambda_B^{N*} = \mathbf{0}$  for the initial iteration. Inserting Eqs. (61), (62), and (63) into (57), (58) yields

$$\begin{aligned} (\mathbf{M}^A + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^A} + \Phi_{\mathbf{q}^A}^N T \alpha \Phi_{\mathbf{q}^A}^N) \dot{\mathbf{q}}^A \\ + \Phi_{\mathbf{q}^A}^T \alpha \Phi_{\mathbf{q}^B} \dot{\mathbf{q}}^B + \Phi_{\mathbf{q}^A}^1 T \lambda_1 \\ = \mathbf{M}^A \dot{\mathbf{q}}_*^A - \Phi_{\mathbf{q}^A}^T (\lambda^* + \alpha \Phi_t^{AB}) - \Phi_{\mathbf{q}^A}^N T \lambda_A^{N*}, \end{aligned} \quad (64)$$

$$\begin{aligned} \Phi_{\mathbf{q}^B}^T \alpha \Phi_{\mathbf{q}^A} \dot{\mathbf{q}}^A + (\mathbf{M}^B + \Phi_{\mathbf{q}^B}^T \alpha \Phi_{\mathbf{q}^B} + \Phi_{\mathbf{q}^B}^N T \alpha \Phi_{\mathbf{q}^B}^N) \dot{\mathbf{q}}^B \\ + \Phi_{\mathbf{q}^B}^2 T \lambda_2 \\ = \mathbf{M}^B \dot{\mathbf{q}}_*^B - \Phi_{\mathbf{q}^B}^T (\lambda^* + \alpha \Phi_t^{AB}) - \Phi_{\mathbf{q}^B}^N T \lambda_B^{N*}. \end{aligned} \quad (65)$$

Equations (64) and (65) form a basis for further derivations. Again, we can write the generalized relations for the system of articulated bodies as shown in Fig. 3, which enable to correct all perturbed velocities:

$$\mathbf{M}_{11}^A \dot{\mathbf{q}}_1^A + \mathbf{M}_{12}^A \dot{\mathbf{q}}_2^A + \Phi_{\mathbf{q}_1^A}^1 T \lambda_1 = \mathbf{Q}_1^A, \quad (66)$$

$$\mathbf{M}_{21}^A \dot{\mathbf{q}}_1^A + \mathbf{M}_{22}^A \dot{\mathbf{q}}_2^A + \Phi_{\mathbf{q}_2^A}^T \lambda = \mathbf{Q}_2^A, \quad (67)$$

and

$$\mathbf{M}_{11}^B \dot{\mathbf{q}}_1^B + \mathbf{M}_{12}^B \dot{\mathbf{q}}_2^B + \Phi_{\mathbf{q}_1^B}^T \lambda = \mathbf{Q}_1^B, \quad (68)$$

$$\mathbf{M}_{21}^B \dot{\mathbf{q}}_1^B + \mathbf{M}_{22}^B \dot{\mathbf{q}}_2^B + \Phi_{\mathbf{q}_2^B}^2 T \lambda_2 = \mathbf{Q}_2^B. \quad (69)$$

The objective of the algorithm for mass-orthogonal projections is to find relations describing articulated body  $C$  (Fig. 3) in the form:

$$\mathbf{M}_{11}^C \dot{\mathbf{q}}_1^C + \mathbf{M}_{12}^C \dot{\mathbf{q}}_2^C + \Phi_{\mathbf{q}_1^C}^1 T \lambda_1 = \mathbf{Q}_1^C, \quad (70)$$

$$\mathbf{M}_{21}^C \dot{\mathbf{q}}_1^C + \mathbf{M}_{22}^C \dot{\mathbf{q}}_2^C + \Phi_{\mathbf{q}_2^C}^2 T \lambda_2 = \mathbf{Q}_2^C. \quad (71)$$

The Lagrange multipliers associated with the joint between sets of bodies  $A$  and  $B$  can be approximated as follows (see Eq. (110) in the Appendix):

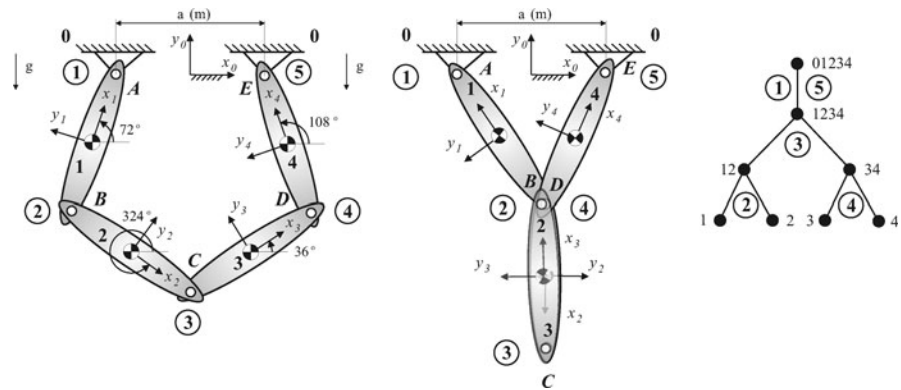
$$\lambda = \lambda^* + \alpha(\Phi_{\mathbf{q}_2^A} \dot{\mathbf{q}}_2^A + \Phi_{\mathbf{q}_1^B} \dot{\mathbf{q}}_1^B + \Phi_t^{AB}). \quad (72)$$

The relations (66)–(72) are of the same structure as Eqs. (11)–(17). The matrix coefficients  $\mathbf{M}_{ij}^A$ ,  $\mathbf{M}_{ij}^B$  for  $i, j = 1, 2$  are computed earlier in the process of projections in positions, whereas vector quantities  $\mathbf{Q}_i^A$ ,  $\mathbf{Q}_i^B$  for  $i = 1, 2$  are evaluated in each iteration as a function of corrected velocities. The projections in velocities are continued until the specified convergence conditions are met, e.g.,  $\|\dot{\mathbf{q}}^{(i+1)} - \dot{\mathbf{q}}^{(i)}\| < \epsilon_v$ , where  $\epsilon_v$  is the user specified tolerance.

The mass-orthogonal projections in positions and velocities can be performed at each time step to obtain a set of values, which satisfy constraints equations within the user specified tolerance. Afterward, the corrected state variables are used to evaluate accelerations according to the divide and conquer scheme described in Sect. 2. The matrix coefficients  $\mathbf{M}_{ij}$  for  $i, j = 1, 2$  from Eqs. (27)–(30) do not need to be reevaluated at this computational stage, because they are computed earlier in the process of position projections. The vector quantities  $\mathbf{Q}_i$  for  $i = 1, 2$  from Eqs. (31), (32) are computed as a function of corrected state variables. If the projections are performed at each time step, the choice of parameters  $\Omega$  and  $\mu$  is irrelevant. They may be set to zero, because the position and velocity constraint equations are satisfied after the projections to the constraint manifold.



**Fig. 4** Five-bar mechanism, its singular configuration, and the binary tree associated with computations



## 4 Numerical examples

### 4.1 Introduction

This section presents some results of the numerical experiments, which are performed to prove the correctness of the developed formulations as well as indicate some of its features. All of the test cases are simulated with intent of gathering characteristics of the parallel divide and conquer algorithm without and with mass-orthogonal projections. The comparisons are made with respect to accuracy and robustness of the formulations. All of the test cases are systems with closed kinematic loops and they are modeled as spatial mechanisms with redundant constraints, which may undergo singular configurations. Apart from the numerical issues, the ways of modeling MBS with closed kinematic chains are demonstrated and generalized.

The test cases are implemented in Matlab in double precision arithmetic, with standard *ode45* integration procedure [40]. The absolute and relative tolerances are set to  $10^{-6}$ . All bodies are modeled as rigid interconnected via revolute joints only. The characteristic length of each body in the system is  $a = 0.2$  m, with masses  $m_i = 1$  kg and inertia matrices equal to  $\mathbf{J}_i' = \text{diag}(0.25)$  kg m<sup>2</sup> with respect to the local reference frames  $(x_i, y_i, z_i)$  with the origin located at the centers of masses of bodies. It is assumed that the penalty factor is set to  $\alpha = \text{diag}(10^6)$ , whereas other matrix coefficients, if needed, are  $\mu = \text{diag}(1.0)$ ,  $\Omega = \text{diag}(1.0)$ . The maximum number of iterations is limited to 10 at position, velocity and acceleration level, and simultaneously user specified tolerances are chosen to be  $\epsilon_p = 10^{-14}$ ,  $\epsilon_v = 10^{-12}$  and  $\epsilon_a = 10^{-10}$ ,

respectively. For each test multibody systems 10 second simulations are performed, with the mechanisms released from the initial state shown in figures under gravity forces.

### 4.2 Single kinematic loop

Consider the five-bar mechanism as shown in Fig. 4. The system is a representative of multibody systems with closed kinematic chains. Bodies 1 and 4 are connected to nonmovable base 0 to create a kinematic loop. All of the joints are revolute, with the axes of revolution perpendicular to the plane of the motion (page). From Grubler's formula, it appears that the mobility of this spatial mechanism is  $w = 6 \cdot 4 - 5 \cdot 5 = -1$ , whereas in fact the system possess two degrees of freedom. Thus, three redundant constraint are imposed on the system. This situation implicates the permanent row rank deficiency of the Jacobian matrix. Moreover, the five-bar mechanism may undergo singular configurations as shown in Fig. 4, at that time the Jacobian matrix temporarily loses its current rank.

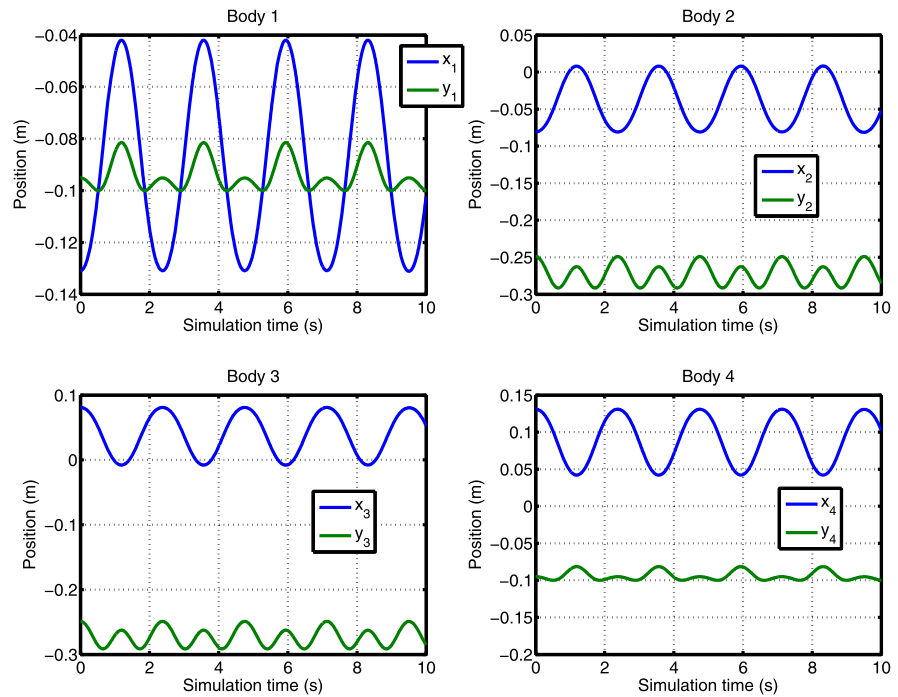
At the initial instant, the Cartesian coordinates of the points associated with axes of revolution in joints are located at the apexes of the pentagon and initial velocities are set to zero. After the assembly phase, the resulting equations of motion associated with the root node of the binary tree depicted in Fig. 4 can be expressed as follows:

$$\mathbf{M}_{11}^{1-4} \ddot{\mathbf{q}}_1 + \mathbf{M}_{12}^{1-4} \ddot{\mathbf{q}}_4 + \Phi_{q_1}^T \lambda_1 = \mathbf{Q}_1^{1-4}, \quad (73)$$

$$\mathbf{M}_{21}^{1-4} \ddot{\mathbf{q}}_1 + \mathbf{M}_{22}^{1-4} \ddot{\mathbf{q}}_4 + \Phi_{q_4}^T \lambda_5 = \mathbf{Q}_2^{1-4}. \quad (74)$$

The Lagrange multipliers  $\lambda_1$  and  $\lambda_5$  correspond to constraint forces in first and fifth joint. These quantities can be approximated as follows:

**Fig. 5** Simulation results for the five-bar mechanism obtained by using DCA algorithm without projections



$$\lambda_1 = \lambda_1^* + \alpha(\Phi_{q_1}^1 \ddot{q}_1 + \bar{y}_{01}), \quad (75)$$

$$\lambda_5 = \lambda_5^* + \alpha(\Phi_{q_4}^5 \ddot{q}_4 + \bar{y}_{04}). \quad (76)$$

Inserting Eqs. (75), (76) into (73), (74), we obtain the linear system of equations with respect to accelerations of bodies 1 and 4:

$$\begin{bmatrix} \mathbf{M}_{11}^{1-4} + \Phi_{q_1}^1 \alpha \Phi_{q_1}^1 & \mathbf{M}_{12}^{1-4} \\ \mathbf{M}_{21}^{1-4} & \mathbf{M}_{22}^{1-4} + \Phi_{q_4}^5 \alpha \Phi_{q_4}^5 \end{bmatrix} \times \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1^{1-4} - \Phi_{q_1}^1 T (\lambda_1^* + \alpha \bar{y}_{01}) \\ \mathbf{Q}_2^{1-4} - \Phi_{q_4}^5 T (\lambda_5^* + \alpha \bar{y}_{04}) \end{bmatrix}. \quad (77)$$

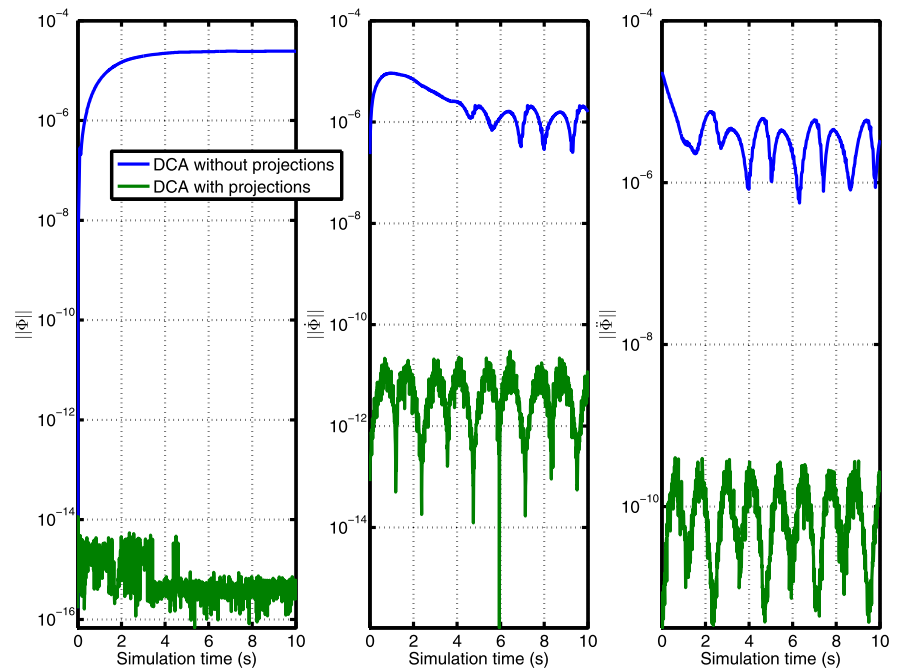
It should be emphasized that the matrix coefficients in Eq. (77) are symmetric and positive definite even in the case, when Jacobian matrices are rank deficient (permanently or temporarily). In addition, there is a relation  $\mathbf{M}_{12}^{1-4} = (\mathbf{M}_{21}^{1-4})^T$ , which can be used for improving computational efficiency. After evaluating  $\ddot{q}_1$ ,  $\ddot{q}_4$  and Lagrange multipliers  $\lambda_1$ ,  $\lambda_5$ , the disassembly phase is started to compute all of bodies' accelerations and constraint forces in joints.

Figure 5 shows the results obtained from a 10 second simulation of the five-bar multibody system when moving under gravity forces. The motion of the mechanism is smooth, without abrupt changes in values of kinematic parameters, even when passing repeatedly through singular configurations. The same qualitative and quantitative results are obtained for DCA algorithm with projections.

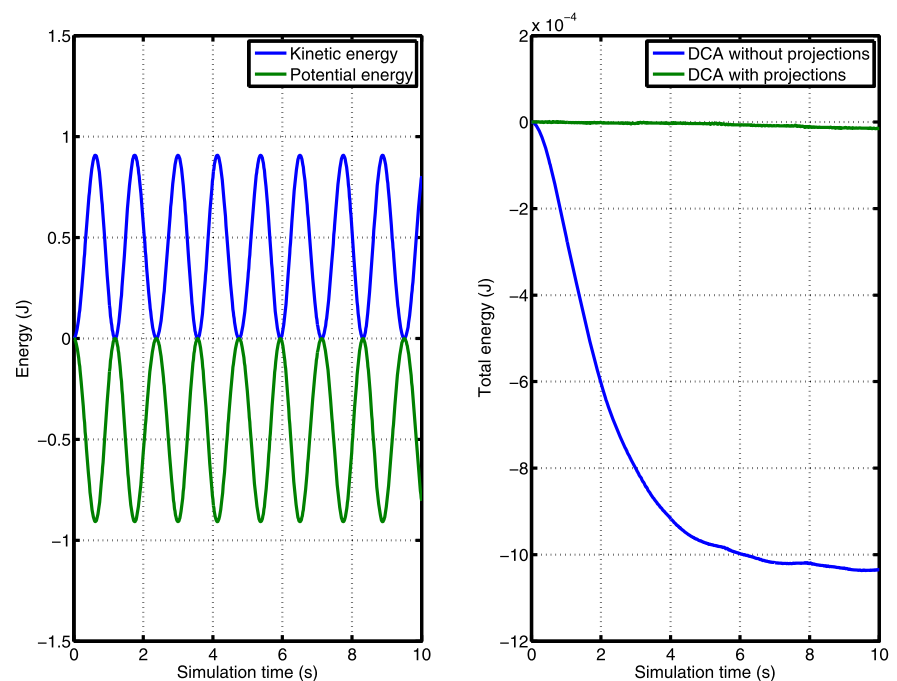
Figure 6 presents the constraint violation errors at position, velocity, and acceleration levels with respect to time. As can be noticed, both formulations assure the stabilization of constraint equations. The DCA algorithm with mass-orthogonal projections, performed each time step, indicates better constraints satisfaction compared to the formulation without projections. However, the improvement in accuracy requires additional computational expense, which refers to the numerical cost of projections in positions and velocities.

The plots of the kinetic, potential, and total energy for the five-bar mechanism are shown in Fig. 7. Similarly, as in case of constraint violations, the DCA algorithm with projections indicates better properties. For this case, the total energy of the system is conserved more accurately compared to the DCA algorithm without projections.

**Fig. 6** Constraint errors in positions, velocities and accelerations for the five-bar mechanism



**Fig. 7** Kinetic, potential, and total energy of the five-bar mechanism



On the basis of obtained results, it can be seen that the proposed algorithms behave well for dynamics simulation of closed loop multibody systems with redundant constraints, which may repeatedly pass through singular configurations. Violations in con-

straint equations are reduced considerably. The formulations demonstrate advantages over traditional methods for constraint imposition based on Lagrange multipliers approach that may fail completely in considered test cases.

### 4.3 Coupled kinematic loops

The approach for dynamics simulation of single kinematic loop systems presented in Sect. 4.2 can be generalized for mechanisms with many coupled kinematic loops. As a simple representative of the group of systems, let us consider the mechanism shown in Fig. 8 with two coupled kinematic loops. Body 3 is common both for upper and lower closed kinematic chains. The theoretical mobility of the mechanism is equal to  $w = 6 \cdot 8 - 5 \cdot 10 = -2$ . In fact the system possesses four degrees of freedom, i.e., six redundant constraints are imposed on the system.

Initially, the Cartesian coordinates of the points associated with axes of revolution in joints are located at apexes of two pentagons coupled together as shown in Fig. 8. The velocities are set to zero at first instant. Let us consider the binary tree depicted in Fig. 8 in order to formulate the equations of motion of the mechanism with coupled loops. At first, the lower kinematic loop is taken into account (bodies 5 to 8). The objective of the transformation is to reduce lower kinematic loop and to couple resulting relations with equations of motion of body 3. After that operation, the upper kinematic loop (bodies 1 to 4) will be considered as a single kinematic loop and its treatment will be similar to that presented in Sect. 4.2. After the assembly phase, the equations of motion of the assembly associated with the lower kinematic loop can be written as

$$\mathbf{M}_{11}^{5-8} \ddot{\mathbf{q}}_5 + \mathbf{M}_{12}^{5-8} \ddot{\mathbf{q}}_8 + \Phi_{q_5}^6 \lambda_6 = \mathbf{Q}_1^{5-8}, \quad (78)$$

$$\mathbf{M}_{21}^{5-8} \ddot{\mathbf{q}}_5 + \mathbf{M}_{22}^{5-8} \ddot{\mathbf{q}}_8 + \Phi_{q_8}^{10} \lambda_{10} = \mathbf{Q}_2^{5-8}. \quad (79)$$

The accelerations of bodies 5 and 8 can be evaluated from the system of linear equations (78) and (79) in terms of constraint forces in joints 6 and 10:

$$\begin{aligned} \begin{bmatrix} \ddot{\mathbf{q}}_5 \\ \ddot{\mathbf{q}}_8 \end{bmatrix} &= \begin{bmatrix} \mathbf{M}_{11}^{5-8} & \mathbf{M}_{12}^{5-8} \\ \mathbf{M}_{21}^{5-8} & \mathbf{M}_{22}^{5-8} \end{bmatrix}^{-1} \\ &\cdot \left\{ \begin{bmatrix} \mathbf{Q}_1^{5-8} \\ \mathbf{Q}_2^{5-8} \end{bmatrix} - \begin{bmatrix} \Phi_{q_5}^6 & \mathbf{0} \\ \mathbf{0} & \Phi_{q_8}^{10} \end{bmatrix}^T \begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} \right\} \\ &= (\mathbf{M})^{-1} \left\{ \mathbf{Q} - \Phi_q^T \begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} \right\}. \end{aligned} \quad (80)$$

On the other hand, the Lagrange multipliers associated with constraint forces in joint 6 and 10 can be approximated as follows:

$$\lambda_6 = \lambda_6^* + \alpha (\Phi_{q_3}^6 \ddot{\mathbf{q}}_3 + \Phi_{q_5}^6 \ddot{\mathbf{q}}_5 + \bar{\gamma}_{35}), \quad (81)$$

$$\lambda_{10} = \lambda_{10}^* + \alpha (\Phi_{q_3}^{10} \ddot{\mathbf{q}}_3 + \Phi_{q_8}^{10} \ddot{\mathbf{q}}_8 + \bar{\gamma}_{38}). \quad (82)$$

Equations (81) and (82) can be rewritten in matrix form:

$$\begin{aligned} \begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} &= \begin{bmatrix} \lambda_6^* \\ \lambda_{10}^* \end{bmatrix} \\ &+ \alpha \left\{ \Phi_q \begin{bmatrix} \ddot{\mathbf{q}}_5 \\ \ddot{\mathbf{q}}_8 \end{bmatrix} + \underbrace{\begin{bmatrix} \Phi_{q_3}^6 \\ \Phi_{q_3}^{10} \end{bmatrix}}_{=\Psi_q} \ddot{\mathbf{q}}_3 + \begin{bmatrix} \bar{\gamma}_{35} \\ \bar{\gamma}_{38} \end{bmatrix} \right\} \\ &= \begin{bmatrix} \lambda_6^* \\ \lambda_{10}^* \end{bmatrix} + \alpha \left\{ \Phi_q \begin{bmatrix} \ddot{\mathbf{q}}_5 \\ \ddot{\mathbf{q}}_8 \end{bmatrix} + \Psi_q \ddot{\mathbf{q}}_3 + \bar{\gamma} \right\}. \end{aligned} \quad (83)$$

Inserting Eq. (80) into (83), we obtain

$$\begin{aligned} &\left( \frac{1}{\alpha} \mathbf{I} + \Phi_q \mathbf{M}^{-1} \Phi_q^T \right) \begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} \\ &= \frac{1}{\alpha} \mathbf{I} \begin{bmatrix} \lambda_6^* \\ \lambda_{10}^* \end{bmatrix} + \Phi_q \mathbf{M}^{-1} \mathbf{Q} + \Psi_q \ddot{\mathbf{q}}_3 + \bar{\gamma} \Rightarrow \\ &\begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} = \mathbf{C} \left( \frac{1}{\alpha} \begin{bmatrix} \lambda_6^* \\ \lambda_{10}^* \end{bmatrix} + \Phi_q \mathbf{M}^{-1} \mathbf{Q} + \Psi_q \ddot{\mathbf{q}}_3 + \bar{\gamma} \right), \end{aligned} \quad (84)$$

where  $\mathbf{C} = \left( \frac{1}{\alpha} \mathbf{I} + \Phi_q \mathbf{M}^{-1} \Phi_q^T \right)^{-1}$  is symmetric and positive definite matrix. It can be seen that Eq. (84) connects Lagrange multipliers  $\lambda_6$ ,  $\lambda_{10}$  with accelerations of body 3. Let us consider equations of motion for body 3:

$$\begin{aligned} \mathbf{M}_3 \ddot{\mathbf{q}}_3 + \Phi_{q_3}^3 \lambda_3 + \Phi_{q_3}^4 \lambda_4 + \Psi_q^T \begin{bmatrix} \lambda_6 \\ \lambda_{10} \end{bmatrix} \\ + \Phi_{q_3}^N \lambda_3^N = \mathbf{Q}_3. \end{aligned} \quad (85)$$

Inserting Eq. (84) into (85), we obtain finally

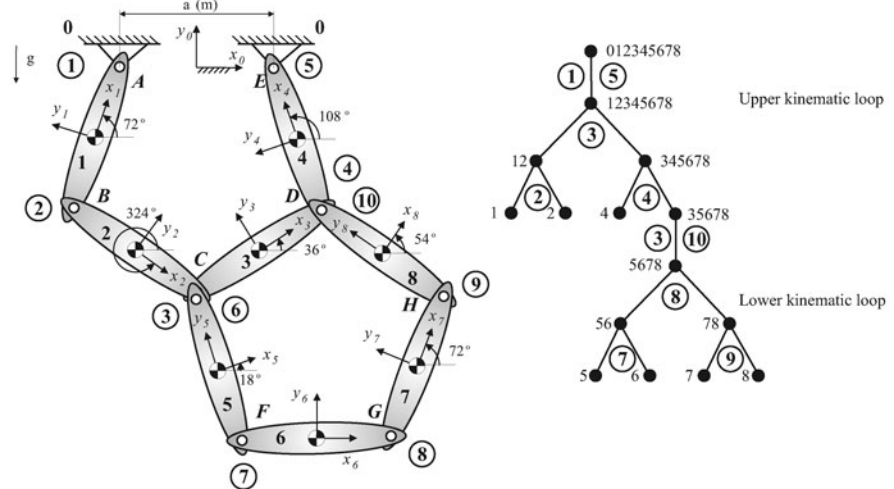
$$\bar{\mathbf{M}}_3 \ddot{\mathbf{q}}_3 + \Phi_{q_3}^3 \lambda_3 + \Phi_{q_3}^4 \lambda_4 + \Phi_{q_3}^N \lambda_3^N = \bar{\mathbf{Q}}_3, \quad (86)$$

where  $\bar{\mathbf{M}}_3 = \mathbf{M}_3 + \Psi_q^T \mathbf{C} \Psi_q$  and

$$\bar{\mathbf{Q}}_3 = \mathbf{Q}_3 - \Psi_q^T \mathbf{C} \left( \frac{1}{\alpha} \begin{bmatrix} \lambda_6^* \\ \lambda_{10}^* \end{bmatrix} + \Phi_q \mathbf{M}^{-1} \mathbf{Q} + \bar{\gamma} \right).$$

It should be emphasized that Eq. (86) expresses dynamics of body 3 as well as motion of lower kinematic loop. The operation of reduction of lower kinematic loop enables to treat upper loop as single closed kinematic chain. Therefore, it is possible to apply the similar formulation as described in Sect. 4.2 to that case.

**Fig. 8** Double loop mechanism and the binary tree associated with computations



**Fig. 9** Simulation results for the double loop mechanism obtained by using DCA algorithm without projections

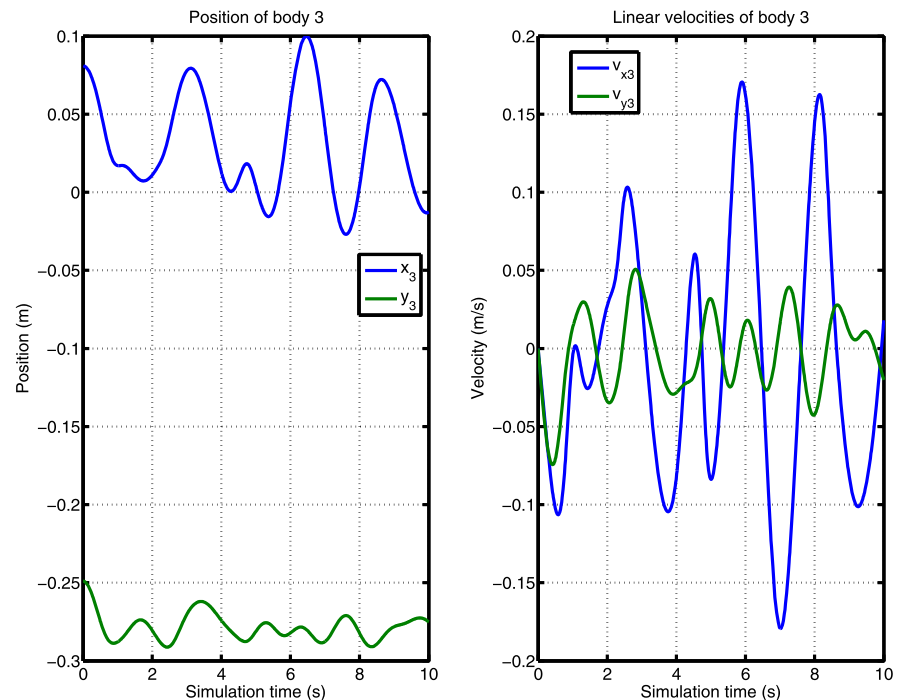


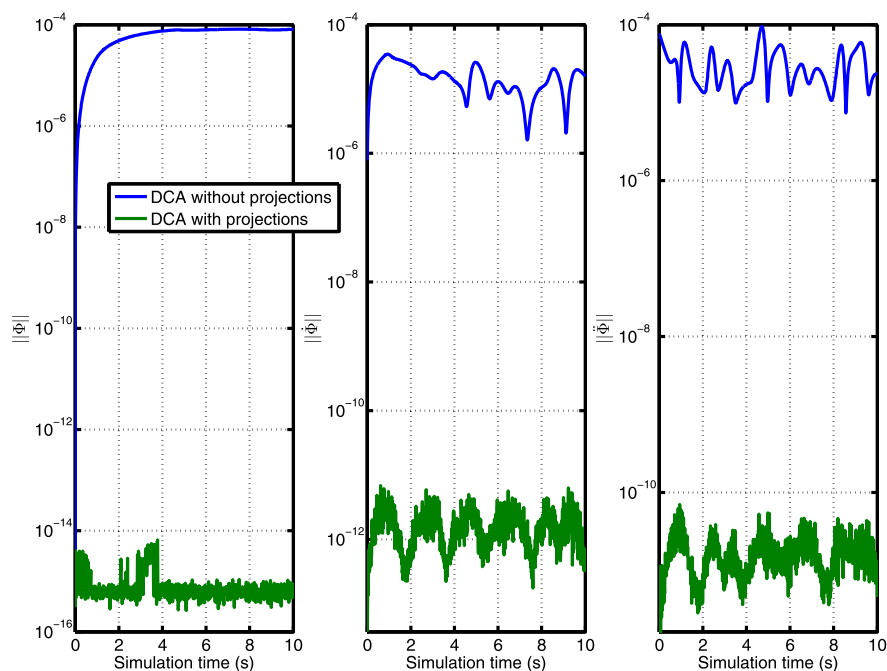
Figure 9 shows the results obtained from dynamics simulation of the double loop mechanism. As in the case of the five-bar mechanism, the motion is smooth without violent changes in kinematic parameters. Constraint errors in positions, velocities, and accelerations shown in Fig. 10 are significantly reduced even up to machine precision in case the DCA algorithm with projections is applied. Also, the total energy of the system depicted in Fig. 11 is well conserved being

superior for the DCA with mass-orthogonal projections.

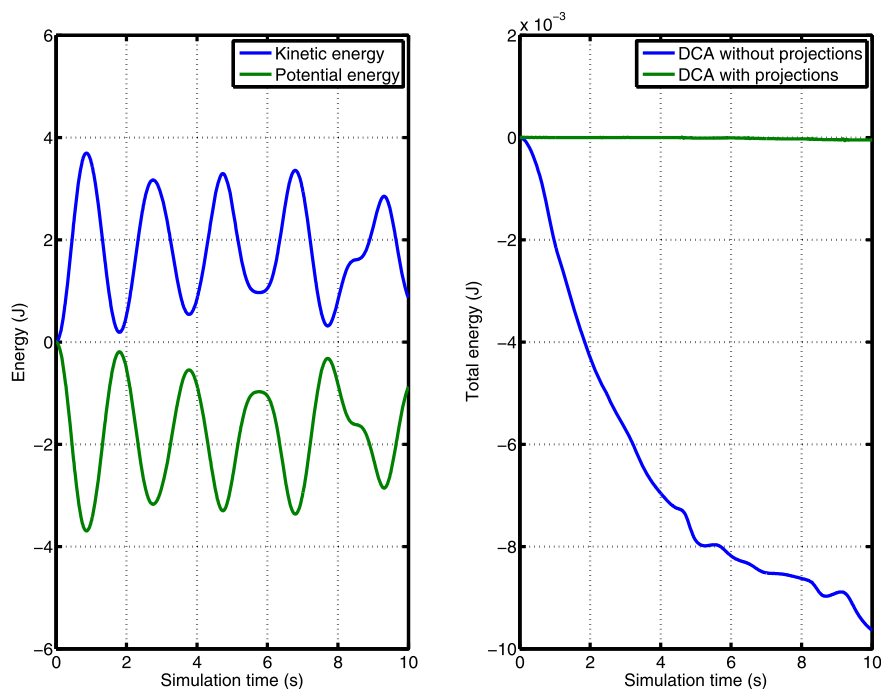
#### 4.4 Discussion

The constraint equations for multibody systems considered in the paper are imposed at the acceleration level. The solution of mixed differential equations of motion together with such algebraic relations suffers from accumulation of constraint errors.

**Fig. 10** Constraint errors in positions, velocities, and accelerations for the double loop mechanism



**Fig. 11** Kinetic, potential, and total energy of the double loop mechanism



Constraint imposition at the acceleration level may lead to substantial violation of the position and velocity constraint equations and, in consequence, unacceptable results for a given simulation. The problem may be solved by introducing constraint stabi-

lization techniques. Special formulations have to be taken into account when dealing with redundant constraints in multibody systems and some peculiarities of the methodologies should be concerned in modeling of mechanisms passing through singular configurations.



The first variant of the formulation presented in the paper combines the divide and conquer algorithm with the augmented Lagrangian method. Lagrange multipliers are approximated by the weighted sum of constraint equations and its two derivatives. The algorithm is robust in the case of modeling systems with redundant constraints, and may handle singular configurations. The second variant maintains all the advantages of the former method and yields better constraints fulfillment by applying mass-orthogonal projections to the constraint manifold at position and velocity level. This algorithm appears to be much more accurate than the regular DCA with augmented Lagrangian method but at the expense of computational burden associated with error corrections.

In the context of DCA based methods, the singularities may be overcome by different approaches. Featherstone in his original work [16] includes the capability of dealing with closed loop systems. For the most general case, the kinematic loops are treated by using generalized inverses and the loop constraint equations are made stable with the add of well-known Baumgarte technique [8]. On the other hand, Mukherjee and Anderson [32] developed a noniterative divide and conquer algorithm for forward dynamics of MBS with single and coupled loops that incorporates orthogonal complements of the joint motion subspace. This method indicates well constraint satisfaction without necessity of applying additional constraint stabilization techniques. In the paper, the singular configurations are overcome by introducing the approximation of Lagrange multipliers based on the augmented Lagrangian method. Due to that fact, the key matrices in the algorithms are nonsingular even when Jacobian matrices lose permanently or exceptionally their rank. The proposed algorithms are inherently iterative. Therefore, some additional computational burden is expected for the first and second variant of the methodology as compared to mentioned counterparts.

The main computational load for the DCA algorithm combined with the augmented Lagrangian formulation is associated with the first iteration. The process of assembly requires the calculation of coefficients according to Eqs. (27)–(32). Taking into account the boundary conditions presented in Sect. 2.3, the disassembly phase is started by exploiting Eqs. (21), (23), and (17). The computational burden is significantly reduced each next iteration, because the matrices  $\mathbf{M}_{ij}$  for  $i, j = 1, 2$  are constant in the iterative process. Vectors  $\mathbf{Q}_i$  for  $i = 1, 2$  and accelerations

are the only quantities that should be reevaluated each iteration. Looking carefully at the recursive formulae, one can find that the efficiency can be further improved by computing only these values, which are dependent on the Lagrange multipliers and accelerations from the previous iteration. This remark allows to avoid expensive matrix multiplications and inversions and, in consequence, diminish the overall numerical cost of the formulation to the extent possible.

The additional computational burden is expected for the second formulation proposed in this work. The added load is associated with the mass-orthogonal projections to the constraint manifold. Similar conclusions may be drawn for the error correction stage at position and velocity level compared to the regular algorithm. The matrix coefficients in the form of  $\mathbf{M}_{ij}$  for  $i, j = 1, 2$  are evaluated at the beginning of the projection at the position level. During the iterative process, they are constant, and then are once reevaluated before the start of the velocity projection scheme. It should be noted that these values remain also constant at the velocity and acceleration level. Vectors  $\mathbf{Q}_i$  for  $i = 1, 2$  and demanded unknowns (positions, velocities, or accelerations) are the only quantities that are altered each iteration. The experience of the authors demonstrated that the penalty factors ranging from  $10^5$  to  $10^7$  give a good convergence rate at each level of computations. The number of iterations is dependent on the user specified tolerances but usually at most 4 iterations are sufficient to obtain reasonable solutions for considered mechanisms.

In conclusion, the computational complexity of both variants considered in the work can be approximated as  $O(kn)$  sequentially and becomes  $O(k \log_2 n)$  for parallel implementations on  $O(n)$  processors, where  $k$  is the number of iterations. A more detailed and quantitative discussion is needed to estimate the numerical cost associated with the proposed algorithms compared to other approaches available in the literature. These issues are areas of ongoing work of the authors and is planned to be addressed in a forthcoming publications.

## 5 Conclusions

In this paper, new formulations for application to general multibody system dynamics have been developed, verified, and compared. Obtained iterative algorithms

are linear, when considered sequentially and achieve logarithmic complexity for parallel computing. Developed methods treat tree topology systems, and general systems which contain kinematic closed loops in a uniform manner and can handle situations, when constraint Jacobian matrices lose its rank. The methods based on augmented Lagrangian formulations with projections ensure appropriate numerical properties. Numerical results of the test cases indicate good accuracy performance dependent on the expense put in the iterative refinement of constraint equations. The formulations are robust in case of analyzing systems with redundant constraints, and which may enter singular configurations. The developed formulations can be extended for analyzing dynamics of complex systems with unilateral constraints, which are areas of forthcoming research for the authors.

**Acknowledgements** This work has been supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme, realized by Center for Advanced Studies.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## Appendix

This Appendix is a supplement to some equations presented in the paper. Mass-orthogonal projections in positions and velocities to the constraints manifold are demonstrated here in a global form. The notation differs slightly from the original one described in [10, 24]. It is adjusted to the notation used in the algorithms presented in the paper.

### 6.1 Mass-orthogonal projections in positions

After the integration time-step, position coordinates  $\mathbf{q}_*$  may not satisfy constraint equations  $\Phi = \mathbf{0}$  with the desired accuracy. In order to minimize these errors, mass-orthogonal projections to the position constraint manifold are performed. Corrected values of  $\mathbf{q}$  can be evaluated by solving the following constrained minimization problem:

$$\min_{\mathbf{q}} V = \min_{\mathbf{q}} \left[ \frac{1}{2} (\mathbf{q} - \mathbf{q}_*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}_*) \right], \quad (87)$$

subjected to nonlinear position constraint equations:

$$\Phi(\mathbf{q}, t) = \mathbf{0}. \quad (88)$$

We will assume that the mass matrix  $\mathbf{M}$  is constant. However, it can be shown that the following procedure is valid also for a nonconstant mass matrix. The minimization problem (87) subjected to equality constraint equations (88) can be solved using classical Lagrange multipliers method [35]. Lagrange function is expressed in the form:

$$\bar{V} = \frac{1}{2} (\mathbf{q} - \mathbf{q}_*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}_*) + \Phi^T \lambda_p, \quad (89)$$

where  $\lambda_p$  are Lagrange multipliers<sup>1</sup> associated with constraint equations  $\Phi = \mathbf{0}$ . The necessary conditions for optimality in constrained problem can be written as

$$\bar{H}(\mathbf{q}, \lambda_p, t) = \left[ \frac{\partial \bar{V}}{\partial \mathbf{q}} \right]^T = \mathbf{M}(\mathbf{q} - \mathbf{q}_*) + \Phi_q^T \lambda_p = \mathbf{0}, \quad (90)$$

$$\left[ \frac{\partial \bar{V}}{\partial \lambda_p} \right]^T = \Phi(\mathbf{q}, t) = \mathbf{0}. \quad (91)$$

Nonlinear set of equations (90) are solved using the Newton–Raphson procedure:

$$\begin{aligned} & \underbrace{\bar{H}(\mathbf{q}^{(i)} + \Delta \mathbf{q}^{(i+1)}, \lambda_p^{(i)} + \Delta \lambda_p^{(i+1)}, t)}_{=0} \\ & \approx \bar{H}(\mathbf{q}^{(i)}, \lambda_p^{(i)}, t) + \bar{H}_q(\mathbf{q}^{(i)}, \lambda_p^{(i)}, t) \Delta \mathbf{q}^{(i+1)} \\ & \quad + \bar{H}_{\lambda_p}(\mathbf{q}^{(i)}, \lambda_p^{(i)}, t) \Delta \lambda_p^{(i+1)}, \end{aligned} \quad (92)$$

where  $\Delta \mathbf{q}^{(i+1)} = \mathbf{q}^{(i+1)} - \mathbf{q}^{(i)}$  and  $\Delta \lambda_p^{(i+1)} = \lambda_p^{(i+1)} - \lambda_p^{(i)}$ . Equations (92) can be expressed in a following form:

$$\mathbf{M} \Delta \mathbf{q}^{(i+1)} + \Phi_q^T \lambda_p^{(i+1)} = -\mathbf{M}(\mathbf{q}^{(i)} - \mathbf{q}_*). \quad (93)$$

Constraint equations (91) are nonlinear in terms of positions  $\mathbf{q}$  and can be solved by applying the iterative scheme:

$$\left[ \frac{\partial \bar{V}}{\partial \lambda_p} \right]^T = \underbrace{\Phi^{(i+1)}}_{=0} \approx \Phi^{(i)} + \Phi_q^{(i)} \Delta \mathbf{q}^{(i+1)}, \quad (94)$$

where  $\Phi^{(i)} = \Phi(\mathbf{q}^{(i)}, t)$ . Taking into account Eq. (93) and Eq. (94) yields:

<sup>1</sup>Subscript  $p$  in Lagrange multipliers  $\lambda_p$  associated with constrained minimization problem (87) and (88) is omitted in Sect. 3.2.

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix}^{(i)} \begin{bmatrix} \Delta \mathbf{q} \\ \lambda_p \end{bmatrix}^{(i+1)} = \begin{bmatrix} -\mathbf{M}(\mathbf{q} - \mathbf{q}_*) \\ -\Phi \end{bmatrix}^{(i)}. \quad (95)$$

The linear set of Eqs. (95) are iterative, however, for the clarity, the indices will be omitted in further derivations. The process starts with perturbed values  $\mathbf{q}^{(i)} = \mathbf{q}_*$  and Lagrange multipliers are set to  $\lambda_p^{(i)} = \mathbf{0}$ .

To minimize functional (87) subjected to constraints (88) augmented Lagrangian method [10, 35] may be used. Then the following unconstrained minimization problem is considered:

$$\min_{\mathbf{q}} V^* = \min_{\mathbf{q}} \left[ \frac{1}{2} (\mathbf{q} - \mathbf{q}_*)^T \mathbf{M} (\mathbf{q} - \mathbf{q}_*) + \frac{1}{2} \Phi^T \alpha \Phi + \Phi^T \lambda_p^* \right], \quad (96)$$

where matrix  $\alpha = \text{diag}(\alpha, \dots, \alpha)$  includes penalty coefficients. The optimality conditions for the unconstrained minimization (96) can be written as

$$H^*(\mathbf{q}, t) = \left[ \frac{\partial V^*}{\partial \mathbf{q}} \right]^T = \mathbf{M}(\mathbf{q} - \mathbf{q}_*) + \Phi_q^T \alpha \Phi + \Phi_q^T \lambda_p^* = \mathbf{0}. \quad (97)$$

Relation (97) can be solved using Newton–Raphson procedure with respect to  $\mathbf{q}$ :

$$\begin{aligned} H^*(\mathbf{q} + \Delta \mathbf{q}, t) &= \mathbf{M}(\mathbf{q} - \mathbf{q}_*) + \Phi_q^T \alpha \Phi + \Phi_q^T \lambda_p^* + \mathbf{M} \Delta \mathbf{q} \\ &\quad + \Phi_{qq}^T \alpha \Phi \Delta \mathbf{q} + \Phi_q^T \alpha \Phi_q \Delta \mathbf{q} + \Phi_{qq}^T \lambda_p \Delta \mathbf{q} \\ &= \mathbf{0}. \end{aligned} \quad (98)$$

The relation (98) includes the term, which is artificially denoted as  $\Phi_{qq}$ . In fact, this is a tensor quantity, which should be written in index notation. It is important to note, for further derivations, that the quantity  $\Phi_{qq}$  is sparse. The components of Eq. (98), which include the quantity  $\Phi_{qq}$  can be neglected, since they are significantly smaller than  $\Phi_q^T \alpha \Phi_q$  [10, 24]. Therefore, Eq. (98) can be expressed as

$$\begin{aligned} (\mathbf{M} + \Phi_q^T \alpha \Phi_q) \Delta \mathbf{q} + \Phi_q^T (\lambda_p^* + \alpha \Phi) \\ = -\mathbf{M}(\mathbf{q} - \mathbf{q}_*). \end{aligned} \quad (99)$$

The leading (tangent) matrix  $\mathbf{M} + \Phi_q^T \alpha \Phi_q$  is symmetric and positive definite (therefore invertible), even in singular positions and/or with redundant constraints. The considered nonlinear problem can be solved efficiently using a modified Newton–Raphson

procedure with constant tangent matrix. Comparing Eq. (99) to Eq. (93) and taking into account Eq. (94), we can set up the following iterative procedure to calculate corrected values of  $\mathbf{q}$  and  $\lambda_p$ :

$$\begin{aligned} \mathbf{q}^{(i+1)} &= \mathbf{q}^{(i)} + \Delta \mathbf{q}^{(i+1)}, \\ \lambda_p^{(i+1)} &= \lambda_p^{(i)} + \alpha \Phi^{(i+1)} \\ &\approx \lambda_p^{(i)} + \alpha (\Phi_q \Delta \mathbf{q}^{(i+1)} + \Phi^{(i)}). \end{aligned} \quad (100)$$

The iterative scheme in Eq. (99) and (100) is continued until the stop criterion is met, e.g.,  $\|\Delta \mathbf{q}\| < \epsilon_p$ , where  $\epsilon_p$  is a user specified tolerance.

## 6.2 Mass-orthogonal projections in velocities

Similarly, the numerical integration procedure yields a set of velocities  $\dot{\mathbf{q}}_*$ , which may not satisfy completely velocity constraint equations  $\dot{\Phi} = \mathbf{0}$ . Again, we perform a mass-orthogonal projections of the solution to velocity constraint manifold. The correction of velocities can be done by the solution of the following constrained minimization problem:

$$\min_{\dot{\mathbf{q}}} V = \min_{\dot{\mathbf{q}}} \left[ \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*)^T \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*) \right], \quad (101)$$

subjected to velocity constraint equations:

$$\dot{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t) = \Phi_q \dot{\mathbf{q}} + \Phi_t = \mathbf{0}. \quad (102)$$

The Lagrange function can be expressed as

$$\bar{V} = \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*)^T \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*) + \dot{\Phi}^T \lambda_v, \quad (103)$$

where  $\lambda_v$  are Lagrange multipliers<sup>2</sup> associated with constraint equations  $\dot{\Phi} = \mathbf{0}$ . The optimality conditions of the considered constrained minimization problem take the form:

$$\begin{aligned} \bar{H}(\mathbf{q}, \dot{\mathbf{q}}, t) &= \left[ \frac{\partial \bar{V}}{\partial \dot{\mathbf{q}}} \right]^T = \mathbf{M}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_*) + \Phi_q^T \lambda_v \\ &= \mathbf{0}, \end{aligned} \quad (104)$$

$$\left[ \frac{\partial \bar{V}}{\partial \lambda_v} \right]^T = \Phi_q \dot{\mathbf{q}} + \Phi_t = \mathbf{0}. \quad (105)$$

Equations (104) and (105) can be expressed in a compact form:

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T \\ \Phi_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \lambda_v \end{bmatrix} = \begin{bmatrix} \mathbf{M} \dot{\mathbf{q}}_* \\ -\Phi_t \end{bmatrix}. \quad (106)$$

<sup>2</sup>In Sect. 3.3, subscript  $v$  in Lagrange multipliers  $\lambda_v$  associated with correction of velocities is omitted.

The augmented Lagrangian method is used again to minimize the functional (101) subjected to equality constraint equations (102):

$$\min_{\dot{\mathbf{q}}} V^* = \min_{\dot{\mathbf{q}}} \left[ \frac{1}{2} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*)^T \mathbf{M} (\dot{\mathbf{q}} - \dot{\mathbf{q}}_*) + \frac{1}{2} \dot{\Phi}^T \alpha \dot{\Phi} + \dot{\Phi}^T \lambda_v^* \right]. \quad (107)$$

The optimality conditions for unconstrained minimization (107) can be written in a form:

$$\begin{aligned} H^*(\mathbf{q}, \dot{\mathbf{q}}, t) &= \left[ \frac{\partial V^*}{\partial \dot{\mathbf{q}}} \right]^T \\ &= \mathbf{M}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_*) + \Phi_{\mathbf{q}}^T \alpha \dot{\Phi} + \Phi_{\mathbf{q}}^T \lambda_v^* \\ &= \mathbf{0}. \end{aligned} \quad (108)$$

Inserting Eq. (102) into Eq. (108) yields:

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \dot{\mathbf{q}} = \mathbf{M} \dot{\mathbf{q}}_* - \Phi_{\mathbf{q}}^T (\lambda_v^* + \alpha \Phi_t). \quad (109)$$

It should be noted that the leading matrix in Eq. (109) is constant during mass-orthogonal projections in velocities. Comparing Eq. (108) to Eq. (104), the following set of relations for Lagrange multipliers  $\lambda_v$  are obtained:

$$\begin{aligned} \lambda_v^{(i+1)} &= \lambda_v^{(i)} + \alpha \dot{\Phi}^{(i+1)} \\ &= \lambda_v^{(i)} + \alpha (\Phi_{\mathbf{q}} \dot{\mathbf{q}}^{(i+1)} + \Phi_t). \end{aligned} \quad (110)$$

The relation (110) can be transformed in order to eliminate Lagrange multipliers  $\lambda_v$ . Applying Eq. (104), we obtain

$$(\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \dot{\mathbf{q}}^{(i+1)} = \mathbf{M} \dot{\mathbf{q}}^{(i)} - \Phi_{\mathbf{q}}^T \alpha \Phi_t. \quad (111)$$

The iterative scheme in Eq. (110) and Eq. (111) is continued until stop criterion is met, e.g.,  $\|\dot{\mathbf{q}}^{(i+1)} - \dot{\mathbf{q}}^{(i)}\| < \epsilon_v$ , where  $\epsilon_v$  is a user specified tolerance. The iterative process starts with perturbed values  $\dot{\mathbf{q}}^{(i)} = \dot{\mathbf{q}}_*$  and Lagrange multipliers are set to  $\lambda_v^{(i)} = \mathbf{0}$ .

## References

1. Anderson, K.: An order  $n$  formulation for the motion simulation of general multi-rigid-body constrained systems. *Comput. Struct.* **3**, 565–579 (1992)
2. Anderson, K., Critchley, J.: Improved ‘order- $n$ ’ performance algorithm for the simulation of constrained multi-rigid-body dynamic systems. *Multibody Syst. Dyn.* **9**, 185–225 (2003)
3. Anderson, K.S., Duan, S.: Highly parallelizable low-order dynamics simulation algorithm for multi-rigid-body systems. *J. Guid. Control Dyn.* **23**, 355–364 (2000)

4. Bae, D.S., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics, part I: open loop systems. *Mech. Struct. Mach.* **15**, 359–382 (1987)
5. Bae, D.S., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics, part II: closed loop systems. *Mech. Struct. Mach.* **15**, 481–506 (1988)
6. Bae, D.S., Kuhl, J., Haug, E.J.: A recursive formulation for constrained mechanical system dynamics, part III: parallel processor implementation. *Mech. Struct. Mach.* **16**, 249–269 (1988)
7. Baraff, D.: Linear-time dynamics using Lagrange multipliers. In: *SIGGRAPH 96, Computer Graphics Proceedings*, New Orleans, pp. 137–146 (1996)
8. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. *Comput. Methods Appl. Mech. Eng.* **1**, 1–16 (1972)
9. Bayo, E., de Jalon, J.G., Serna, M.: A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. *Comput. Methods Appl. Mech. Eng.* **71**, 183–195 (1988)
10. Bayo, E., Ledesma, R.: Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dyn.* **9**, 113–130 (1996)
11. Blajer, W.: Augmented Lagrangian formulation: geometrical interpretation and application to systems with singularities and redundancy. *Multibody Syst. Dyn.* **8**, 141–159 (2002)
12. Chung, S., Haug, E.J.: Real-time simulation of multibody dynamics on shared memory multiprocessors. *J. Dyn. Syst. Meas. Control* **115**, 627–637 (1993)
13. Critchley, J., Anderson, K.: A parallel logarithmic order algorithm for general multibody system dynamics. *Multibody Syst. Dyn.* **12**, 75–93 (2004)
14. Featherstone, R.: The calculation of robot dynamics using articulated-body inertias. *Int. J. Robot. Res.* **2**, 13–30 (1983)
15. Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel  $O(\log n)$  calculation of rigid body dynamics, part 1: basic algorithm. *Int. J. Robot. Res.* **18**, 867–875 (1999)
16. Featherstone, R.: A divide-and-conquer articulated body algorithm for parallel  $O(\log n)$  calculation of rigid body dynamics. Part 2: trees, loops, and accuracy. *Int. J. Robot. Res.* **18**, 876–892 (1999)
17. Featherstone, R.: *Rigid Body Dynamics Algorithms*. Springer, Berlin (2008)
18. Fijany, A., Bejczy, A.K.: Techniques for parallel computation of mechanical manipulator dynamics, part II: forward dynamics. *Control Dyn. Syst.* **40**, 357–410 (1991)
19. Fijany, A., Sharf, I., D’Eleuterio, G.: Parallel  $O(\log n)$  algorithms for computation of manipulator forward dynamics. *IEEE Trans. Robot. Autom.* **11**, 389–400 (1995)
20. Fiset, P., Peterkenne, J.M.: Contribution to parallel and vector computation in multibody dynamics. *Parallel Comput.* **24**, 717–728 (1998)
21. Haug, E.J.: *Computer-Aided Kinematics and Dynamics of Mechanical Systems. Vol. I: Basic Methods*. Allyn & Bacon, Needham Heights (1989)
22. Jain, A.: Unified formulation of dynamics for serial rigid multibody systems. *J. Guid. Control Dyn.* **14**, 531–542 (1991)

23. Jain, A.: Robot and Multibody Dynamics: Analysis and Algorithms, 1st edn. Springer, Berlin (2011)
24. de Jalon, G., Bayo, E.: Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge. Springer, Berlin (1994)
25. Kasahara, H., Fujii, H., Iwata, M.: Parallel processing of robot motion simulation. In: Proceedings IFAC World Congress, Munich (1987)
26. Lee, C., Chang, P.: Efficient parallel algorithms for robot forward dynamics. *IEEE Trans. Syst. Man Cybern.* **18**(2), 238–251 (1988)
27. Malczyk, P.: Modeling of spatial mechanisms using parallel computing. PhD Thesis, Warsaw University of Technology, Poland (2010) (in Polish)
28. Malczyk, P., Frączek, J.: Lagrange multipliers based divide and conquer algorithm for dynamics of general multibody systems. In: Arczewski, K., Frączek, J., Wojtyra, M. (eds.) Proceedings of the ECCOMAS Multibody Dynamics 2009 Thematic Conference, Warsaw, Poland (2009) (CD-ROM). ISBN 978-83-7207-813-1
29. Malczyk, P., Frączek, J.: Parallel efficiency of Lagrange multipliers based divide and conquer algorithm for dynamics of multibody systems. In: Proceedings of the ASME 2011 International Design Engineering Technical Conferences, Washington, DC (2011)
30. Malczyk, P., Frączek, J., Cuadrado, J.: Parallel index-3 formulation for real-time multibody dynamics simulations. In: Proceedings of the First Joint International Conference on Multibody System Dynamics, IMSD 2010, Lappeenranta, Finland (2010). ISBN 978-952-214-778-3
31. Malczyk, P., Frączek, J., Tomulik, P.: A parallel algorithm for constrained multibody system dynamics based on augmented Lagrangian method with projections. In: Proceedings of the ECCOMAS Multibody Dynamics 2011 Thematic Conference, Brussels, Belgium (2011)
32. Mukherjee, R., Anderson, K.: Orthogonal complement based divide-and-conquer algorithm for constrained multibody systems. *Nonlinear Dyn.* **48**, 199–215 (2007)
33. Naudet, J.: Forward dynamics of multibody systems: a recursive Hamiltonian approach. PhD Thesis, Universytet Vrije, Brussels, Belgium (2005)
34. Nikravesh, P.E.: Computer-Aided Analysis of Mechanical Systems. Prentice Hall, New York (1988)
35. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Berlin (2006)
36. Rodriguez, G.: Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE J. Robot. Autom.* **6**, 624–639 (1987)
37. Rosenthal, D.: An order  $n$  formulation for robotic systems. *J. Astronaut. Sci.* **38**(4), 511–529 (1990)
38. Saha, K.S., Schiehlen, W.: Recursive kinematics and dynamics for parallel structured closed-loop multibody systems. *Mech. Struct. Mach.* **29**(2), 143–175 (2001)
39. Stejskal, V., Valasek, M.: Kinematics and Dynamics of Machinery. Marcel Dekker, NY (1996)
40. MATLAB: Online documentation for mathworks products, r2010a. Available online at <http://www.mathworks.com> (2010)
41. Walker, M.W., Orin, D.E.: Efficient dynamic computer simulation of robotic mechanisms. *ASME J. Dyn. Syst. Meas. Control* **104**, 205–211 (1982)
42. Yamane, K., Nakamura, Y.:  $O(N)$  forward dynamics computation of open kinematic chains based on the principle of virtual work. In: Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea, pp. 2824–2831 (2001)
43. Yamane, K., Nakamura, Y.: Parallel  $O(\log n)$  algorithm for dynamics simulation of humanoid robots. In: Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, pp. 554–559 (2006)
44. Yamane, K., Nakamura, Y.: Comparative study on serial and parallel forward dynamics algorithms for kinematic chains. *Int. J. Robot. Res.* **28**(5), 622–629 (2009)