The 1st Joint International Conference on Multibody System Dynamics
May 25–27, 2010, Lappeenranta, Finland

# Parallel index-3 formulation for real-time multibody dynamics simulations

**Paweł Malczyk**[*], **Janusz Frączek**[*], **Javier Cuadrado**[#]

[*] 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, jfraczek@meil.pw.edu.pl

[#] Escuela Politécnica Superior
University of La Coruña
Mendizábal s/n, 15403 Ferrol, Spain
e-mail: javicuad@cdf.udc.es

## ABSTRACT

Computational efficiency of the dynamics of large constrained multibody systems is essential in many areas of computer aided engineering and design. There are many examples of such systems in various disciplines. The involved applications include robots, vehicles, biomechanical systems and other interdisciplinary systems. Moreover, some applications like hardware-in-the-loop or human-in-the-loop devices are developed by applying real-time simulations. Parallel computing is one of the approaches to increase the computational efficiency of multibody simulations. This paper presents a new parallel formulation for real-time multibody dynamics simulations. The proposed method adopts a divide and conquer scheme. Initially, the parallel algorithm is formulated by using classical index-3 Lagrangian approach with trapezoidal rule as a numerical integrator. Subsequently, index-3 augmented Lagrangian formulations with projections are adopted to generalize the algorithm and make it more robust in case of Jacobian matrix rank deficiencies, which may occur in analysing complex systems. The performance results indiciate that real-time computations can be achieved for multibody systems with 128 DOF for three processors used, keeping the constraint errors under control.

**Keywords:** divide and conquer algorithm, augmented Lagrangian formulation with projections, real-time simulation, parallel computing.

## 1 INTRODUCTION

Multibody dynamics simulations 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 earliest examples of recursive sequential algorithms for analysis of rigid body dynamics can be found in [1], [3], [4], [13], [20], [26] and [27]. They gave the basis for further development of efficient low order formulations. As parallel computing resources became more available, researchers began to parallelize 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 [5], [7], [17], [18], [21]. More recent ideas regarding parallel algorithms for rigid multibody dynamics simulation come from Featherstone and Anderson's works. Featherstone [14], [15] and [16] developed the optimal-time 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 [19] and independently Anderson and Duan [2] adopted the idea of a decomposition of multibody system into subchains. In 2004, Critchley and Anderson [8] explored 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 (i.e. matrices relating dependent state variables to independent ones). In 2007, Mukherjee and Anderson [24] presented exact and non-iterative divide and conquer algorithm for forward dynamics of MBS with single and coupled loops. The formulation incorporated neither coordinate reductions nor Lagrange multipliers. The constraint equations were imposed at the acceleration level through kinematic relations involving orthogonal complement of the joint motion subspace. The algorithm indicated good constraint fulfillment and could easily handle systems in singular configurations.

In a number of large industrial tasks there is a need to achieve real-time performance. The group of applications encompasses not only hardware-in-the-loop or human-in-the-loop computations but also the simulation of complex, realistic interdisciplinary systems from area of e.g. biomechanics, robotics, aerospace, heavy machinery and military industries. Much work was done in this field, which is evident in [6], [12]. The specialized formulations were developed to meet real-time performance conditions. The most efficient one appears to be index-3 formulation with projections worked out by Cuadrado et al. [9], [10] and [11]. Apart from the efficiency of the formulation, the index-3 formulation is more robust than classical Lagrangian formulations. It does not cause failures in case of singular configurations and redundant constraints often encountered in analysis of realistic multibody systems. Moreover the accuracy, in terms of constraint error violations, is kept under control of the user, which is a strong advantage, when simulating complex, large multibody systems.

This paper presents a new parallel algorithm for real-time dynamics simulation of general multi-rigid-body systems. The idea of the method comes from [22], however the authors verified that the formulation was inaccurate, which was a direct consequence of ill-conditioning of associated matrices. In the proposed method, we avoid the problem by careful algebraic manipulations. The algorithm is formulated by using index-3 augmented Lagrangian formulation with projections [9], [10], where the trapezoidal rule has been embedded into the solution process as an integration rule. The formulation makes use of a divide and conquer scheme, which is similar to that in [14]. The proposed approach significantly reduces the turnaround times for the systems with large number of bodies and can be regarded as a tool for achieving efficient computations for full dynamic multibody models.

## 2 OVERVIEW OF INDEX-3 FORMULATIONS

### 2.1 Classical formulation

This section presents an overview of the index-3 formulation in a global form, which is similar to that presented in [10]. It will be useful in understanding and deriving parallel algorithm in position, veloctity and acceleration level in section 3. Moreover some similarities will be shown between both formulations. Let us write the equations of motion of a multibody system in a form of index-3 differential-algebraic equations

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^{T}\boldsymbol{\lambda} = \boldsymbol{Q}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{1}$$

$$\boldsymbol{\Phi}(\boldsymbol{q}, t) = \boldsymbol{0} \tag{2}$$

where $\boldsymbol{M}$ is the mass matrix, $\ddot{\boldsymbol{q}}$ are the accelerations, $\boldsymbol{\Phi}_{\boldsymbol{q}}$ the Jacobian matrix of the constraint equations, $\boldsymbol{\Phi}$ the constraint vector, $\boldsymbol{\lambda}$ the Lagrange multipliers, and $\boldsymbol{Q}$ the vector of applied and velocity dependent intertia forces. The single-step trapezoidal rule has been adopted as an integration scheme. The difference equations in velocities and accelerations are following:

$$\dot{\boldsymbol{q}}_{n+1} = \frac{2}{\Delta t}\boldsymbol{q}_{n+1} + \hat{\dot{\boldsymbol{q}}}_n \quad \text{where} \quad \hat{\dot{\boldsymbol{q}}}_n = -\Big(\frac{2}{\Delta t}\boldsymbol{q}_n + \dot{\boldsymbol{q}}_n\Big) \tag{3}$$

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{4}{\Delta t^2}\boldsymbol{q}_{n+1} + \hat{\ddot{\boldsymbol{q}}}_n \quad \text{where} \quad \hat{\ddot{\boldsymbol{q}}}_n = -\Big(\frac{4}{\Delta t^2}\boldsymbol{q}_n + \frac{4}{\Delta t}\dot{\boldsymbol{q}}_n + \ddot{\boldsymbol{q}}_n\Big) \tag{4}$$

Equations (3) and (4) are introduced into (1) and (2) at time step $n+1$ yielding

$$\frac{4}{\Delta t^2}\boldsymbol{M}\boldsymbol{q}_{n+1} + \boldsymbol{\Phi}_{\boldsymbol{q}_{n+1}}^{T}\boldsymbol{\lambda}_{n+1} - \boldsymbol{Q}_{n+1} + \boldsymbol{M}\hat{\ddot{\boldsymbol{q}}}_n = \boldsymbol{0} \tag{5}$$

$$\boldsymbol{\Phi}_{n+1} = \mathbf{0} \tag{6}$$

These equations are scaled by a factor of $\frac{\Delta t^2}{4}$ yielding

$$\boldsymbol{M}\boldsymbol{q}_{n+1} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{n+1}}^T\boldsymbol{\lambda}_{n+1} - \frac{\Delta t^2}{4}\boldsymbol{Q}_{n+1} + \frac{\Delta t^2}{4}\boldsymbol{M}\hat{\ddot{\boldsymbol{q}}}_n = \mathbf{0} \tag{7}$$

$$\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{n+1} = \mathbf{0} \tag{8}$$

Equations (7) and (8) form a system of nonlinear equations in $\bar{\boldsymbol{q}} = [\boldsymbol{q}^T, \boldsymbol{\lambda}^T]^T$. These equations may be solved by the Newton-Raphson procedure as in (9):

$$\frac{\partial \boldsymbol{f}(\bar{\boldsymbol{q}}^i)}{\partial \bar{\boldsymbol{q}}}\Delta\bar{\boldsymbol{q}}^{i+1} = -\boldsymbol{f}(\bar{\boldsymbol{q}}^i) \quad \text{where} \quad \Delta\bar{\boldsymbol{q}}^{i+1} = \begin{bmatrix} \Delta\boldsymbol{q}^{i+1} \\ \Delta\boldsymbol{\lambda}^{i+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}^{i+1} - \boldsymbol{q}^i \\ \boldsymbol{\lambda}^{i+1} - \boldsymbol{\lambda}^i \end{bmatrix} \tag{9}$$

where

$$\boldsymbol{f}(\bar{\boldsymbol{q}}) = \begin{bmatrix} \frac{\Delta t^2}{4}(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda} - \boldsymbol{Q}) \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{10}$$

and the tangent matrix can be approximated as

$$\frac{\partial \boldsymbol{f}(\bar{\boldsymbol{q}})}{\partial \bar{\boldsymbol{q}}} = \begin{bmatrix} \boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K} & \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}} & \mathbf{0} \end{bmatrix} \tag{11}$$

where $\boldsymbol{C} = -\frac{\partial \boldsymbol{Q}}{\partial \dot{\boldsymbol{q}}}$ and $\boldsymbol{K} = -\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{q}}$. The equations of motion at the position level are following

$$\begin{bmatrix} \boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K} & \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}} & \mathbf{0} \end{bmatrix}^i \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\lambda} \end{bmatrix}^{i+1} = -\begin{bmatrix} \frac{\Delta t^2}{4}(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda} - \boldsymbol{Q}) \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi} \end{bmatrix}^i \tag{12}$$

The first and second derivatives of constraint equations will not be satisfied during the simulation course, because the constraint conditions have been imposed in positions only. In order to achieve constraint fulfillment, mass-orthogonal projections in velocities and accelerations can be introduced, such that the leading matrices are identical to the tangent matrix (11). If $\dot{\boldsymbol{q}}^*$ and $\ddot{\boldsymbol{q}}^*$ are the velocities and accelerations obtained after convergence of Newton-Raphson procedure (12), the set of new variables that satisfy constraint equations $\dot{\boldsymbol{\Phi}}$ and $\ddot{\boldsymbol{\Phi}}$ can be evaluated once per integration step as follows

$$\begin{bmatrix} \boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K} & \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{\sigma} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K})\dot{\boldsymbol{q}}^* \\ -\frac{\Delta t^2}{4}\boldsymbol{\Phi}_t \end{bmatrix} \tag{13}$$

and

$$\begin{bmatrix} \boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K} & \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T \\ \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\kappa} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K})\ddot{\boldsymbol{q}}^* \\ -\frac{\Delta t^2}{4}\boldsymbol{\gamma} \end{bmatrix} \tag{14}$$

It is worth noting that the leading matrix of the previous systems of equations becomes singular when the Jacobian matrix of the constraints $\boldsymbol{\Phi}_{\boldsymbol{q}}$ does. The situation may happen due to the redundant constraint or singular positions encountered in the multibody system.

### 2.2 Formulation with projections

The leading matrix of the formulation described in section 2.1 is not robust when singular positions and redundant constraints are encountered during the simulation. It is possible to rewrite equations of motion (1) by using index-three formulation, in which the positions $\boldsymbol{q}$ become the primary variables [6], [10]. The corresponding equations of motion are given as

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T(\boldsymbol{\lambda} + \alpha\boldsymbol{\Phi}) = \boldsymbol{Q}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{15}$$

where $\alpha$ is the penalty vector. The Lagrange multipliers are obtained from the following iteration process:

$$\boldsymbol{\lambda}^{i+1} = \boldsymbol{\lambda}^i + \alpha\boldsymbol{\Phi}^{i+1} \quad \text{where} \quad i = 0, 1, 2, \cdots \tag{16}$$

The value $\boldsymbol{\lambda}_0 = 0$ is chosen for the first iteration. The Lagrange multipliers in (16) can be expanded using Taylor series

$$\Delta\boldsymbol{\lambda}^{i+1} = \alpha\boldsymbol{\Phi}^{i+1} \approx \alpha(\boldsymbol{\Phi_q}\Delta\boldsymbol{q}^{i+1} + \boldsymbol{\Phi}^i) \tag{17}$$

Inserting (17) into (12)

$$(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}(\boldsymbol{K} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\Phi_q}))^i\Delta\boldsymbol{q}^{i+1} = -\frac{\Delta t^2}{4}(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T(\boldsymbol{\lambda} + \alpha\boldsymbol{\Phi}) - \boldsymbol{Q})^i \tag{18}$$

The leading matrix is symmetric and positive-definite, thus it is robust in case of singular configurations and redundant constraints, however it reveals that ill-conditioning may appear for small time steps [10]. It is not expected that the corresponding sets of velocities and accelerations satisfy $\dot{\boldsymbol{\Phi}} = \boldsymbol{0}$ and $\ddot{\boldsymbol{\Phi}} = \boldsymbol{0}$, because these conditions have not been imposed in the solution process. Mass-orthogonal projections in velocities and accelerations are introduced to prevent from constraint violations. Lagrange multipliers at the velocity and acceleration level can be expressed respectively:

$$\boldsymbol{\sigma}^{i+1} = \boldsymbol{\sigma}^i + \alpha\dot{\boldsymbol{\Phi}}^{i+1} = \boldsymbol{\sigma}^i + \alpha(\boldsymbol{\Phi_q}\dot{\boldsymbol{q}}^{i+1} + \boldsymbol{\Phi}_t) \tag{19}$$

and

$$\boldsymbol{\kappa}^{i+1} = \boldsymbol{\kappa}^i + \alpha\ddot{\boldsymbol{\Phi}}^{i+1} = \boldsymbol{\kappa}^i + \alpha(\boldsymbol{\Phi_q}\ddot{\boldsymbol{q}}^{i+1} + \boldsymbol{\gamma}) \tag{20}$$

If $\dot{\boldsymbol{q}}^*$ and $\ddot{\boldsymbol{q}}^*$ are the velocities and accelerations obtained from (3) and (4), after convergence of Newton-Raphson procedure (18), the set of new variables that satisfy constraint equations $\dot{\boldsymbol{\Phi}}$ and $\ddot{\boldsymbol{\Phi}}$ can be evaluated by substituting equation (19) and (20) into (13) and (14).

$$(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}(\boldsymbol{K} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\Phi_q}))\dot{\boldsymbol{q}} = (\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K})\dot{\boldsymbol{q}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\Phi}_t \tag{21}$$

and

$$(\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}(\boldsymbol{K} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\Phi_q}))\ddot{\boldsymbol{q}} = (\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K})\ddot{\boldsymbol{q}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\gamma} \tag{22}$$

Now, the leading matrix $\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}(\boldsymbol{K} + \boldsymbol{\Phi}_{\boldsymbol{q}}^T\alpha\boldsymbol{\Phi_q})$ is symmetric and positive definite, even when Jacobian matrices lose their ranks. The convergence of the iteration process defined in (18) can be improved by introducing a predictor. A good second order predictor is the modified trapezoidal rule, which gives the following approximation [12]:

$$\tilde{\boldsymbol{q}}_{n+1}^0 = \boldsymbol{q}_n + \dot{\boldsymbol{q}}_n\Delta t + \ddot{\boldsymbol{q}}_n\frac{\Delta t^2}{2} \tag{23}$$

Instead of using solution from time $n$ as the initial conditions for the next time step, we can take the approximation as in (23). The process is computationally inexpensive but significantly improves the convergence rate.

## 3 INDEX-3 DIVIDE AND CONQUER LAGRANGIAN FORMULATION WITH PROJECTIONS

This section presents analytical derivations for the index-3 parallel algorithm for real-time dynamics simulations of general rigid multibody systems. The system is modeled by the use of absolute coordinates. The modified trapezoidal rule is embedded into the solution process. The proposed method makes use of a divide and conquer algorithm at position, velocity and acceleration level.

Figure 1 depicts a system of two rigid bodies connected by a joint. Each body is characterized by generalized coordinates $\boldsymbol{q}$, which are in the form of $7\times1$ matrix of position and orientation described in Euler parameters $\boldsymbol{q} = [\boldsymbol{r}^T \quad \boldsymbol{p}^T]$. The quantities are associated with the mass center of a body. Equations of motion for bodies $A$ and $B$ with embedded trapezoidal rule can be written as in (12):

$$\widetilde{\boldsymbol{M}}^A\Delta\boldsymbol{q}^A + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{1\,T}\Delta\boldsymbol{\lambda}_1 + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^A}^T\Delta\boldsymbol{\lambda} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N\,T}\Delta\boldsymbol{\lambda}_A^N = \widetilde{\boldsymbol{Q}}^A \tag{24}$$

$$\widetilde{\boldsymbol{M}}^B\Delta\boldsymbol{q}^B + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{2\,T}\Delta\boldsymbol{\lambda}_2 + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^B}^T\Delta\boldsymbol{\lambda} + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N\,T}\Delta\boldsymbol{\lambda}_B^N = \widetilde{\boldsymbol{Q}}^B \tag{25}$$

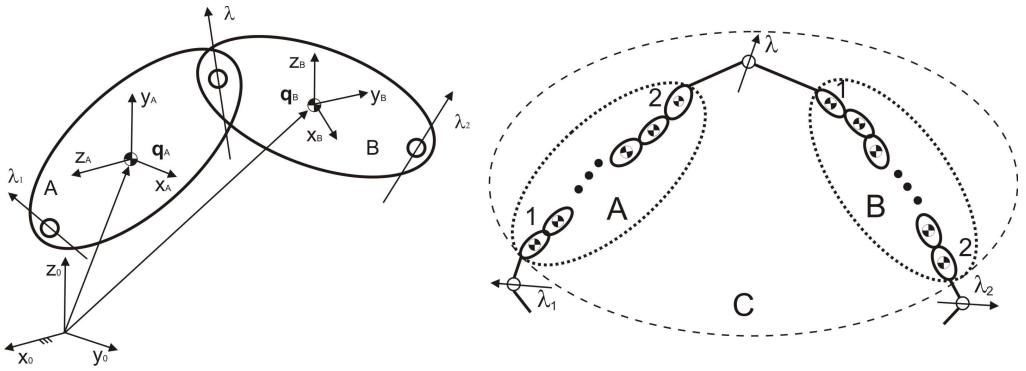


**Figure 1**. System of articulated bodies

where

$$\widetilde{\boldsymbol{M}}^A = \boldsymbol{M}^A + \frac{\Delta t}{2}\boldsymbol{C}^A + \frac{\Delta t^2}{4}\boldsymbol{K}^A \tag{26}$$

$$\widetilde{\boldsymbol{Q}}^A = -\frac{\Delta t^2}{4}(\boldsymbol{M}^A\ddot{\boldsymbol{q}}^A + \boldsymbol{\Phi}_{\boldsymbol{q}^A}^{1\,T}\boldsymbol{\lambda}_1 + \boldsymbol{\Phi}_{\boldsymbol{q}^A}^{T}\boldsymbol{\lambda} + \boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N\,T}\boldsymbol{\lambda}_A^N - \boldsymbol{Q}^A) \tag{27}$$

$$\widetilde{\boldsymbol{M}}^B = \boldsymbol{M}^B + \frac{\Delta t}{2}\boldsymbol{C}^B + \frac{\Delta t^2}{4}\boldsymbol{K}^B \tag{28}$$

$$\widetilde{\boldsymbol{Q}}^B = -\frac{\Delta t^2}{4}(\boldsymbol{M}^B\ddot{\boldsymbol{q}}^B + \boldsymbol{\Phi}_{\boldsymbol{q}^B}^{2\,T}\boldsymbol{\lambda}_2 + \boldsymbol{\Phi}_{\boldsymbol{q}^B}^{T}\boldsymbol{\lambda} + \boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N\,T}\boldsymbol{\lambda}_B^N - \boldsymbol{Q}^B) \tag{29}$$

The leading matrices $\widetilde{\boldsymbol{M}}^A$ and $\widetilde{\boldsymbol{M}}^B$ are not invertible and may cause failures in case of rank deficiency of Jacobian matrices. We introduce Lagrange multipiliers approximations, as in equation (17).

$$\Delta\boldsymbol{\lambda} = \alpha\boldsymbol{\Phi}(\boldsymbol{q}^A, \boldsymbol{q}^B, t) = \alpha(\boldsymbol{\Phi}_{\boldsymbol{q}^A}\Delta\boldsymbol{q}^A + \boldsymbol{\Phi}_{\boldsymbol{q}^B}\Delta\boldsymbol{q}^B + \boldsymbol{\Phi}(\boldsymbol{q}^A, \boldsymbol{q}^B, t)) \tag{30}$$

$$\Delta\boldsymbol{\lambda}_A^N = \alpha\boldsymbol{\Phi}^N(\boldsymbol{q}^A) = \alpha(\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N\,T}\Delta\boldsymbol{q}^A + \boldsymbol{\Phi}^N(\boldsymbol{q}^A)) \tag{31}$$

$$\Delta\boldsymbol{\lambda}_B^N = \alpha\boldsymbol{\Phi}^N(\boldsymbol{q}^B) = \alpha(\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N\,T}\Delta\boldsymbol{q}^B + \boldsymbol{\Phi}^N(\boldsymbol{q}^B)) \tag{32}$$

Substituting relations (30), (31) and (32) into (24) and (25) yields

$$\begin{aligned}&[\widetilde{\boldsymbol{M}}^A + \frac{\Delta t^2}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^A} + \boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N\,T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N})]\Delta\boldsymbol{q}^A + \frac{\Delta t^2}{4}[\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^B}]\Delta\boldsymbol{q}^B + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{1\,T}\Delta\boldsymbol{\lambda}_1 = \\ &= \widetilde{\boldsymbol{Q}}^A - \frac{\Delta t^2}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}^A}^{T}\alpha\boldsymbol{\Phi}(\boldsymbol{q}^A, \boldsymbol{q}^B, t) + \boldsymbol{\Phi}_{\boldsymbol{q}^A}^{N\,T}\alpha\boldsymbol{\Phi}^N(\boldsymbol{q}^A))\end{aligned} \tag{33}$$

$$\begin{aligned}&\frac{\Delta t^2}{4}[\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^A}]\Delta\boldsymbol{q}^A + [\widetilde{\boldsymbol{M}}^B + \frac{\Delta t^2}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^B} + \boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N\,T}\alpha\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N})]\Delta\boldsymbol{q}^B + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{2\,T}\Delta\boldsymbol{\lambda}_2 = \\ &= \widetilde{\boldsymbol{Q}}^B - \frac{\Delta t^2}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}^B}^{T}\alpha\boldsymbol{\Phi}(\boldsymbol{q}^A, \boldsymbol{q}^B, t) + \boldsymbol{\Phi}_{\boldsymbol{q}^B}^{N\,T}\alpha\boldsymbol{\Phi}^N(\boldsymbol{q}^B))\end{aligned} \tag{34}$$

Equations (33) and (34) form a basis for further derivations. They are starting point for the main pass phase, which will be presented below. In contrast to equations (24) and (25), relations (33) and (34) can be solved for positions $\boldsymbol{q}^A$ and $\boldsymbol{q}^B$ respectively. The encountered coefficient matrices are symmetric positive definite, even in situations, when constraint Jacobians lose their ranks. Regarding Figure 1 and expressions (33) and (34), we can generalize formulation to incorporate more than two articulated bodies. Let us consider the system of relations:

$$\boldsymbol{M}_{11}^A\Delta\boldsymbol{q}_1^A + \boldsymbol{M}_{12}^A\Delta\boldsymbol{q}_2^A + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_1^A}^{1\,T}\Delta\boldsymbol{\lambda}_1 = \boldsymbol{Q}_1^A \tag{35}$$

$$\boldsymbol{M}_{21}^A\Delta\boldsymbol{q}_1^A + \boldsymbol{M}_{22}^A\Delta\boldsymbol{q}_2^A + \frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}^{T}\Delta\boldsymbol{\lambda} = \boldsymbol{Q}_2^A \tag{36}$$

$$\boldsymbol{M}_{11}^{B}\Delta\boldsymbol{q}_{1}^{B}+\boldsymbol{M}_{12}^{B}\Delta\boldsymbol{q}_{2}^{B}+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T}\Delta\boldsymbol{\lambda}=\boldsymbol{Q}_{1}^{B} \tag{37}$$

$$\boldsymbol{M}_{21}^{B}\Delta\boldsymbol{q}_{1}^{B}+\boldsymbol{M}_{22}^{B}\Delta\boldsymbol{q}_{2}^{B}+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{B}}^{2\,T}\Delta\boldsymbol{\lambda}_{2}=\boldsymbol{Q}_{2}^{B} \tag{38}$$

The objective of the following algebraic manipulations is to obtain the equations for the set $C$ from the relation (35) to (38):

$$\boldsymbol{M}_{11}^{C}\Delta\boldsymbol{q}_{1}^{A}+\boldsymbol{M}_{12}^{C}\Delta\boldsymbol{q}_{2}^{B}+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{A}}^{1\,T}\Delta\boldsymbol{\lambda}_{1}=\boldsymbol{Q}_{1}^{C} \tag{39}$$

$$\boldsymbol{M}_{21}^{C}\Delta\boldsymbol{q}_{1}^{A}+\boldsymbol{M}_{22}^{C}\Delta\boldsymbol{q}_{2}^{B}+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{B}}^{2\,T}\Delta\boldsymbol{\lambda}_{2}=\boldsymbol{Q}_{2}^{C} \tag{40}$$

Unknown Lagrange multipliers can be found through the following iterative process (see (17)):

$$\Delta\boldsymbol{\lambda}=\alpha\boldsymbol{\Phi}(\boldsymbol{q}_{2}^{A},\boldsymbol{q}_{1}^{B},t)=\alpha(\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}\Delta\boldsymbol{q}_{2}^{A}+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}\Delta\boldsymbol{q}_{1}^{B}+\boldsymbol{\Phi}^{i}) \tag{41}$$

Dividing (41) by penalty factor $\alpha$ and inserting relations (36) and (37) to (41) yields

$$\begin{aligned}\frac{1}{\alpha}\boldsymbol{I}\Delta\boldsymbol{\lambda}=\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}(\boldsymbol{Q}_{2}^{A}-\boldsymbol{M}_{21}^{A}\Delta\boldsymbol{q}_{1}^{A}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}\Delta\boldsymbol{\lambda})+\\+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}(\boldsymbol{Q}_{1}^{B}-\boldsymbol{M}_{12}^{B}\Delta\boldsymbol{q}_{2}^{B}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T}\Delta\boldsymbol{\lambda})+\boldsymbol{\Phi}^{i}\end{aligned} \tag{42}$$

where $\boldsymbol{I}$ is the identity matrix of proper dimension. The relation (42) can be solved in terms of Lagrange multipliers between the set of bodies $A$ and $B$:

$$\begin{aligned}&[\frac{1}{\alpha}\boldsymbol{I}+\frac{\Delta t^{2}}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T})]\Delta\boldsymbol{\lambda}=\\&=\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}(\boldsymbol{Q}_{2}^{A}-\boldsymbol{M}_{21}^{A}\Delta\boldsymbol{q}_{1}^{A})+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}(\boldsymbol{Q}_{1}^{B}-\boldsymbol{M}_{12}^{B}\Delta\boldsymbol{q}_{2}^{B})+\boldsymbol{\Phi}\end{aligned} \tag{43}$$

We denote matrix $\boldsymbol{C}$ as

$$\boldsymbol{C}=[\frac{1}{\alpha}\boldsymbol{I}+\frac{\Delta t^{2}}{4}(\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T})]^{-1} \tag{44}$$

There is no problem with matrix inversion in (44), because of the presence of small values $\frac{1}{\alpha}$, added on the diagonal, which improve the matrix conditioning. Substituting equation (43) into (36) and taking into account (44) we obtain the following relations:

$$\begin{aligned}\Delta\boldsymbol{q}_{2}^{A}=(\boldsymbol{M}_{22}^{A})^{-1}\{\overline{\boldsymbol{Q}}_{2}^{A}-(\boldsymbol{I}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1})\boldsymbol{M}_{21}^{A}\Delta\boldsymbol{q}_{1}^{A}+\\+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{M}_{12}^{B}\Delta\boldsymbol{q}_{2}^{B}\}\end{aligned} \tag{45}$$

where

$$\overline{\boldsymbol{Q}}_{2}^{A}=\boldsymbol{Q}_{2}^{A}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}^{T}\boldsymbol{C}(\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{Q}_{2}^{A}+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{Q}_{1}^{B}+\boldsymbol{\Phi}) \tag{46}$$

We can also insert equation (43) into (37)

$$\begin{aligned}\Delta\boldsymbol{q}_{1}^{B}=(\boldsymbol{M}_{11}^{B})^{-1}\{\overline{\boldsymbol{Q}}_{1}^{B}-(\boldsymbol{I}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1})\boldsymbol{M}_{12}^{B}\Delta\boldsymbol{q}_{2}^{B}+\\+\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{M}_{21}^{A}\Delta\boldsymbol{q}_{1}^{A}\}\end{aligned} \tag{47}$$

where

$$\overline{\boldsymbol{Q}}_{1}^{B}=\boldsymbol{Q}_{1}^{B}-\frac{\Delta t^{2}}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}^{T}\boldsymbol{C}(\boldsymbol{\Phi}_{\boldsymbol{q}_{2}^{A}}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{Q}_{2}^{A}+\boldsymbol{\Phi}_{\boldsymbol{q}_{1}^{B}}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{Q}_{1}^{B}+\boldsymbol{\Phi}) \tag{48}$$

The final step is to substitute positions from equations (45) and (47) into equations (35) and (38) respectively.

$$\begin{aligned}&[\boldsymbol{M}_{11}^{A}-\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}(\boldsymbol{I}-\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}(\boldsymbol{M}_{22}^{A})^{-1})\boldsymbol{M}_{21}^{A}]\Delta\boldsymbol{q}_1^A+\\ &+[\frac{\Delta t^2}{4}\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{M}_{12}^{B}]\Delta\boldsymbol{q}_2^B+\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_1^A}^{1\,T}\Delta\boldsymbol{\lambda}_1=\\ &=\boldsymbol{Q}_1^A-\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}\overline{\boldsymbol{Q}}_2^A\end{aligned}\tag{49}$$

$$\begin{aligned}&[\frac{\Delta t^2}{4}\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{M}_{21}^{A}]\Delta\boldsymbol{q}_1^A+\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_2^B}^{2\,T}\Delta\boldsymbol{\lambda}_2\\ &+[\boldsymbol{M}_{22}^{B}-\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}(\boldsymbol{I}-\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}(\boldsymbol{M}_{11}^{B})^{-1})\boldsymbol{M}_{12}^{B}]\Delta\boldsymbol{q}_2^B=\\ &=\boldsymbol{Q}_2^B-\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}\overline{\boldsymbol{Q}}_1^B\end{aligned}\tag{50}$$

By gathering appropriate matrix coefficients and comparing equations (39) and (40) together with (49) and (50), the following final relations are obtained:

$$\boldsymbol{M}_{11}^{C}=\boldsymbol{M}_{11}^{A}-\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}(\boldsymbol{I}-\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}(\boldsymbol{M}_{22}^{A})^{-1})\boldsymbol{M}_{21}^{A}\tag{51}$$

$$\boldsymbol{M}_{12}^{C}=\frac{\Delta t^2}{4}\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{M}_{12}^{B}\tag{52}$$

$$\boldsymbol{M}_{21}^{C}=\frac{\Delta t^2}{4}\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_2^A}(\boldsymbol{M}_{22}^{A})^{-1}\boldsymbol{M}_{21}^{A}\equiv(\boldsymbol{M}_{12}^{C})^{T}\tag{53}$$

$$\boldsymbol{M}_{22}^{C}=\boldsymbol{M}_{22}^{B}-\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}(\boldsymbol{I}-\frac{\Delta t^2}{4}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}^{T}\boldsymbol{C}\boldsymbol{\Phi}_{\boldsymbol{q}_1^B}(\boldsymbol{M}_{11}^{B})^{-1})\boldsymbol{M}_{12}^{B}\tag{54}$$

and

$$\boldsymbol{Q}_1^C=\boldsymbol{Q}_1^A-\boldsymbol{M}_{12}^{A}(\boldsymbol{M}_{22}^{A})^{-1}\overline{\boldsymbol{Q}}_2^A\tag{55}$$

$$\boldsymbol{Q}_2^C=\boldsymbol{Q}_2^B-\boldsymbol{M}_{21}^{B}(\boldsymbol{M}_{11}^{B})^{-1}\overline{\boldsymbol{Q}}_1^B\tag{56}$$

Equations (51)-(56) define the recursive relations for the object achieved after the binary assembly of the child-objects. When the root node is reached, the equations are similar to (39) and (40). For open chain system $\Delta\boldsymbol{\lambda}_2=\boldsymbol{0}$, which indicate free floating terminal body. In turn, Lagrange multipliers associated with a joint connected to fixed base body can be approximated from (41). We can evaluate positions $\Delta\boldsymbol{q}_1^A$ and $\Delta\boldsymbol{q}_2^B$ from the system of linear equations. Starting with evaluated base and terminal body positions, all positions and Lagrange multipliers for the next time instant can be computed from equations (45), (47) and (41). The divide and conquer scheme in absolute coordinates with embedded trapezoidal rule consists of two passes, the main pass, and the back-substitution pass. The presented algebraic manipulations enable to construct equations of motion of a multibody system by the process of recursive binary assembly and disassembly, as depicted in Figure 2. The process starts with equations of motion for each body in the system (24) and (25). Subassemblies, which correspond to nodes in the graph, are constructed by traversing the binary tree. Finally, the entire MBS is obtained, which is indicated as a root node in 2. The main pass finishes at this stage. Now, taking into account a connection of a chain to fixed base and free floating terminal body, the back-substitution phase is started. All Lagrange multipliers and bodies' positions can be computed in the system, traversing the tree from root node to leaves. Derived parallel algorithm posses the same features as original index-3 formulation does. However the formulation does not enforce simultaneously the errors in velocities and acceleration constraints to the desired level. In order to overcome these drawbacks, mass-orthogonal projections in velocities and accelerations are employed for further considerations as in (13) and (14) together with (19) and (20). These relations are of the same form as in case of (24) and (25), therefore we can apply again the divide and conquer approach described in this paper at the velocity and acceleration level. The projections will be made only once per integration step.

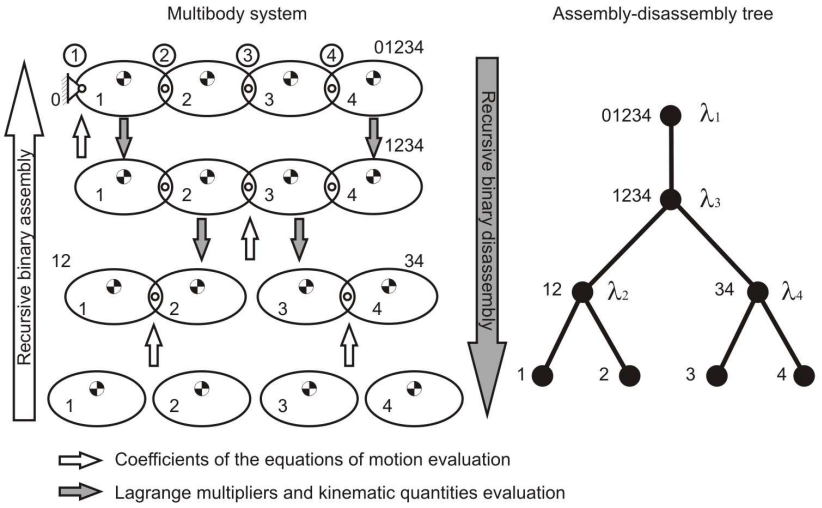


**Figure 2**. Recursive binary assembly and disassembly of a four-link multibody system

## 4 NUMERICAL EXPERIMENTS

### 4.1 Open chain system

This section presents some results of the numerical experiments, which are performed to prove the correctness of the algorithm as well as indicate some of its features. In order to investigate constraint fulfillment of the derived algorithm, a test mechanical system with sixteen degrees of freedom have been created, as shown in Figure 3. Each body in the open chain is connected to each other by simple revolute joints. All

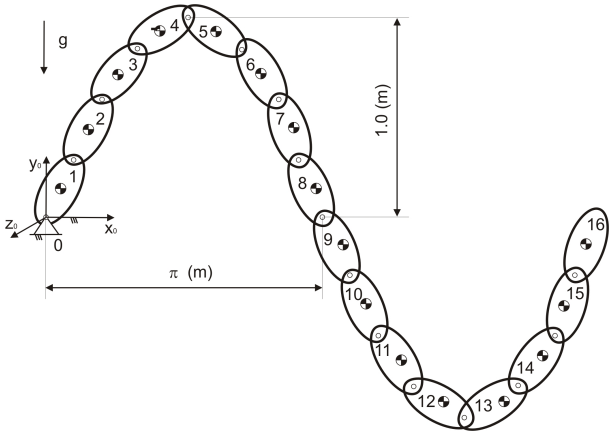


**Figure 3**. A 16 body test mechanical system

axes of revolution are parallel to the global axis $z$. The numbering of the bodies increases successively from the non-movable base body 0 to the terminal body 16. The characteristic points and body mass centres were located on the sine function, as indicated in Figure 3. The properties of each body in the chain are following: mass $m_i = 1kg$, product of inertia expressed as a $3 \times 3$ diagonal matrix $\boldsymbol{I}_i = diag(1.0)kgm^2$, with respect to the reference frames fixed at each body. At initial instant, absolute coordinates of the bodies are nonzero and absolute velocities are set to zero. Distance between the reference points and the body mass center projected onto the global $x$ axis is constant and equals to $\frac{\pi}{16}$. The multibody system is released from non-equilibrium state under gravity forces. The results of the 20-second dynamics simulation, with $\alpha = 10^7$, integration step $\Delta t = 0.01$ second and the stop criterion for the Newton-Raphson procedure taken

as $||\Delta q|| < \epsilon = 10^{-6}$ are depicted in Figure 4.

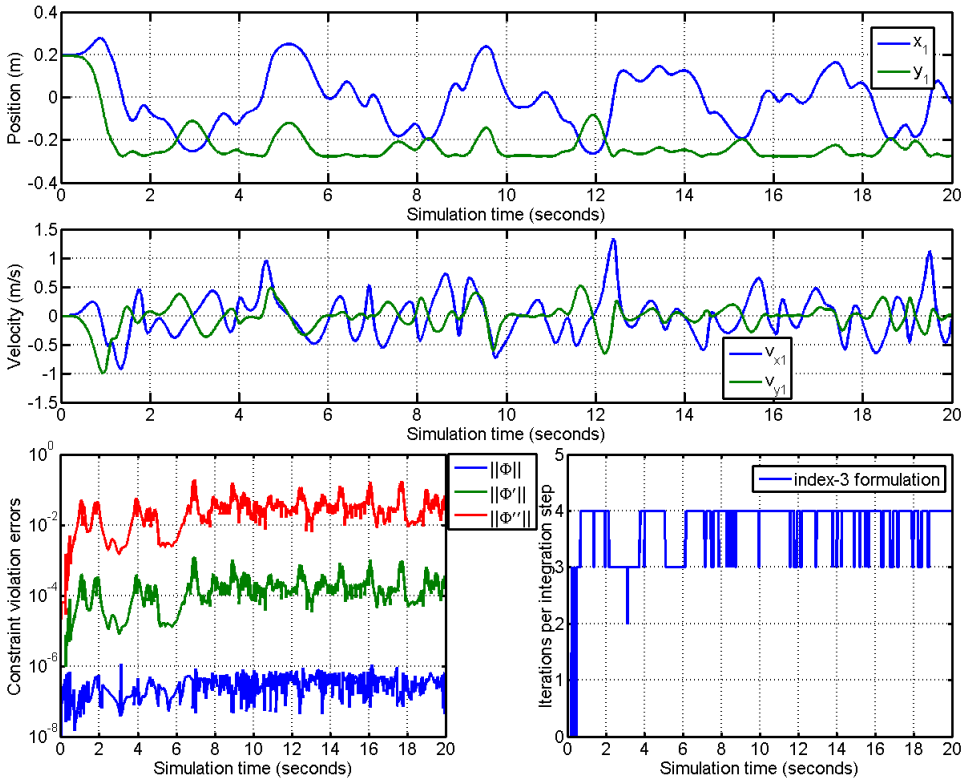


**Figure 4**. Constraint violation errors for 16-body open chain system

The proposed formulation clearly stabilizes the constraint equations at the position, velocity and acceleration levels. Errors in position constraint violations are small compared to the characteristic dimension of the sample multibody system and regarding the defined tolerances. Analysis of the constraint violation errors indicates that the smallest errors are made at the position level. The constraints errors in $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are worse, but still acceptable. The result is a direct consequence of applying trapezoidal rule for the approximation of velocities and accelerations. In Figure 4 we can also observe the number of iterations in terms of simulation time. Few iterations are required to convergence the solutions. When we tighten the $\epsilon$, more computational burden can be expected to fulfill user specified tolerances. The projections in velocities and accelerations are made only once, which is an improvement in performance over the algorithms described in [22], which took into account augmented Lagrangian formulation in a full form without embedding the integration rule.

### 4.2 Performance results

Another issue, which should be considered, is the performance characteristics of the formulation and realistic implementation of the algorithm on the parallel computer. The presented implementation can currently work with spatial open chain systems. It is constructed in such a way that can help to gather the performance results of the dynamic simulation when different multibody systems are considered. The properties of the divide and conquer algorithm are exploited to build the parallel code. According to Figure 2, every node in the system can be computed in parallel at the specified level. The implementation is built upon this idea and consists of two nested loops. The outermost loop consists of $log_2 n$ iterations, whereas the innermost loop traverses through the number of bodies at the indicated level. The amount of concurrency varies over the execution of a program, which can constrain the speedup, especially for low number of bodies in the system. At some level of recursion, the amount of computation may become so small that it is worth grouping tasks and treating them as a one computational subproblem. This approach should be more effective for low num-

ber of bodies compared to the strategy applied in this implementation. The formulations described in this paper are parallelized according to the OpenMP [23], [25] compiler directives and library functions, chosen with respect to their simplicity and portability across shared memory architectures. There are other ways to parallelize the code, e.g. POSIX threads, which are often used in real-time operating systems practice. The POSIX threads add more control over the parallelization and synchronization process and therefore may give better performance results compared to OpenMP directives. The aforementioned innermost loops are parallelized by the use of parallel loop constructs that specify iterations of one or more loops to be executed in parallel by threads in the team. The parallel multibody codes are executed on shared memory parallel computer (Sun Server), running OpenSuse 11.2 Operating System. The parallel computer is equipped with two-socket motherboard, in which two quad-core processors are installed. Each processor is a Quad-Core AMD Opteron Processor 2356 (2.3GHz) with 512kB cache L2 per core and four 2GB ECC DDR-667 memory modules are set in the motherboard. The source codes are compiled with -O3 optimization flag using gfortran 4.3 compiler. For performance evaluation purposes, a test mechanical system with revolute joints is used, as described in previous section. The only change is that the system has adjustable number of bodies. When more bodies are added to the system, the chain is getting longer but the links maintain the same dimension. The simulation time is $10.0$ seconds and the integration time step is set to $0.0175$ second and chosen carefully to ensure stability and convergence along whole simulation time. User specified tolerance is chosen to $\epsilon = 3 \cdot 10^{-4}$ and penalty factor is $\alpha = 10^6$. The maximal number of iterations is limited to four iterations per integration step. The efficiency results of the developed index-3 formulation are compared to timing results obtained from execution of index-1 parallel algorithm presented in [22], where accelerations are primary variables and constraint violation errors at position, velocity and acceleration levels are cleaned with mass-orthogonal projections. The conditions of the numerical experiments are the same in both cases, however the equations of motion evaluated from the cited algorithm are integrated with Runge-Kutta $4^{th}$ order routine. The execution times for different number of bodies may be seen in Figure 5.

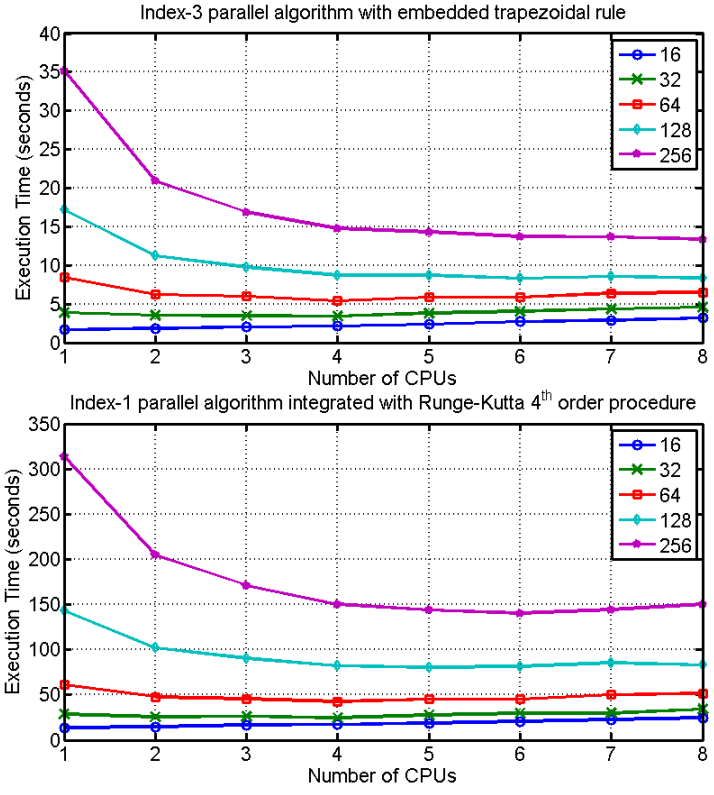


**Figure 5**. Comparison of performance results for parallel index-3 formulation

Figure 5 depicts the timing results obtained during a 10-second dynamics simulation of multibody open chain. The influence of number of degrees of freedom and number of processing elements on the timing

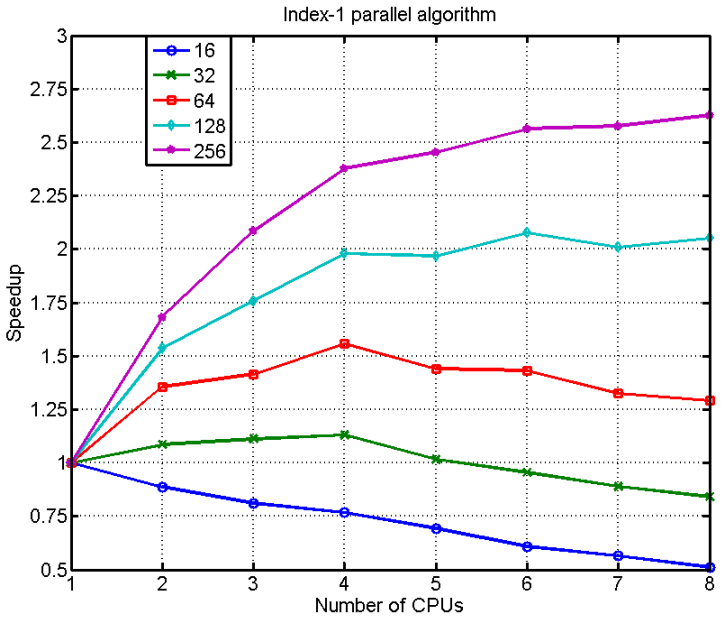


**Figure 6**. Speedup for parallel index-3 formulation

results are considered. None of the simulations performed by parallel algorithm with mass-orthogonal projections [22] meet real-time conditions. Having parallel computer described in this section, when multibody systems with less than 64 bodies are considered, the execution times show good level of efficiency and make moderate use of parallel computing. This effect was observed in cited work and is evident in Fig. 6. For 128 bodies, real-time efficieny can't be approached unless three processors are employed for the computations. It should be noted that the cost associated with thread management reduces useful speed increases for small multibody systems. However, the timing results indicate that the parallelization is more effective in case of large multibody systems, where higher increase in computational load/thread may be expected. In summary, it is clearly seen that the combination of the parallel index-3 formulation with integration scheme gives much better performance results compared to the cited index-1 algorithm. The chosen integration step constitutes a compromise between the levels of accuracy and efficiency. The CPU-savings could be even greater if large time-steps would be allowable for the price of losing the information about dynamics of a multibody system. When large number of bodies are analyzed, the process of solving equations of motion can be numerically stiff. To obtain stable and efficient dynamics simulations for relatively large time-steps, the integrator (e.g. Newmark method) may provide some numerical damping [11]. Similarly, as in the paper, we can add the integration scheme into the formulation to obtain planned characteristics.

## 5 CONCLUSIONS

In the paper, we have developed new parallel, index-3 algorithm for the dynamics of general multibody systems, which encompasses parallel efficiency, acceptable accuracy and robustness. The method is a combination of divide and conquer algorithm together with index-3 approach. Trapezoidal rule has been embedded into the solution process, to ensure good level of efficiency comapared to more standard augmented Lagrangian formulation with mass-orthogonal projections. The method can handle situations, when constraint Jacobian matrices lose its rank, because index-3 approach ensures symmetric positive definite leading matrices. The improvement in efficiency has been observed compared to the previous work. For larger systems the method experiences significant computational benefits from parallel computing and can be regarded as a tool for achieving efficient dynamics simulation for full multibody models.

## ACKNOWLEDGEMENTS

The work was supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme and by the Institute of Aeronautics and Applied Mechanics statutory funds.

The work was co-financed by the European Regional Development Fund within the framework of the 1. priority axis of the Innovative Economy Operational Programme, 2007-2013 - PROTEUS under grant number PO IG.01.02.01-00-014/08-00.

Research co-financed by the Polish Ministry of Science and Higher Education in years 2008-2010 under grant number N N514 236437.

## REFERENCES

[1] ANDERSON, K. An order n formulation for the motion simulation of general multi-rigid-body constrained systems. *Computer and Structures 3* (1992), 565–579.

[2] ANDERSON, K. S., AND DUAN, S. Highly parallelizable low-order dynamics simulation algorithm for multi-rigid-body systems. *Journal of Guidance, Control, and Dynamics 23* (2000), 355–364.

[3] BAE, D. S., AND HAUG, E. J. A recursive formulation for constrained mechanical system dynamics: Part i: Open loop systems. *Mechanics of Structures and Machines 15* (1987), 359–382.

[4] BAE, D. S., AND HAUG, E. J. A recursive formulation for constrained mechanical system dynamics: Part ii: Closed loop systems. *Mechanics of Structures and Machines 15* (1988), 481–506.

[5] BAE, D. S., AND HAUG, E. J. A recursive formulation for constrained mechanical system dynamics: Part iii. parallel processor implementation. *Mechanics of Structures and Machines 16* (1988), 249–269.

[6] BAYO, E., AND LEDESMA, R. Augmented lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dynamics 9* (1996), 113–130.

[7] CHUNG, S., AND HAUG, E. J. Real-time simulation of multibody dynamics on shared memory multiprocessors. *Journal of Dynamic Systems, Measurement, and Control 115* (1993), 627–637.

[8] CRITCHLEY, J., AND ANDERSON, K. A parallel logarithmic order algorithm for general multibody system dynamics. *Multibody System Dynamics 12* (2004), 75–93.

[9] CUADRADO, J., CARDENAL, J., AND BAYO, E. Modeling and solution methods for efficient real-time simulation of multibody dynamics. *Multibody System Dynamics 1* (1997), 259–280.

[10] CUADRADO, J., CARDENAL, J., MORER, P., AND BAYO, E. Intelligent simulation of multibody dynamics: Space-state and descriptor methods in sequential and parallel computing environments. *Multibody System Dynamics 4* (2000), 55–73.

[11] CUADRADO, J., DOPICO, D., NAYA, M., AND GONZALEZ, M. Penalty, semi-recursive and hybrid methods for mbs real-time dynamics in the context of structural integrators. *Multibody System Dynamics 12* (2004), 117–132.

[12] DE JALON, G., AND BAYO, E. *Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challange*. Springer Verlag, Berlin, 1994.

[13] FEATHERSTONE, R. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research 2* (1983), 13–30.

[14] FEATHERSTONE, R. A divide-and-conquer articulated body algorithm for parallel calculation of rigid body dynamics. part 1: Basic algorithm. *International Journal of Robotics Research 18* (1999), 867–875.

[15] FEATHERSTONE, R. A divide-and-conquer articulated body algorithm for parallel calculation of rigid body dynamics. part 2: Trees, loops, and accuracy. *International Journal of Robotics Research 18* (1999), 876–892.

[16] FEATHERSTONE, R. *Rigid Body Dynamics Algorithms*. Springer, 2008.

[17] FIJANY, A., AND BEJCZY, A. K. Techniques for parallel computation of mechanical manipulator dynamics. part ii: Forward dynamics. *Control and Dynamics Systems 40* (1991), 357–410.

[18] FIJANY, A., SHARF, I., AND D'ELEUTERIO, G. Parallel algorithms for computation of manipulator forward dynamics. *IEEE Transactions on Robotics and Automation 11* (1995), 389–400.

[19] FISETTE, P., AND PETERKENNE, J. M. Contribution to parallel and vector computation in multibody dynamics. *Parallel Computing 24* (1998), 717–728.

[20] JAIN, A. Unified formulation of dynamics for serial rigid multibody systems. *Journal of Guidance, Control, and Dynamics 14* (1991), 531–542.

[21] KASAHARA, H., FUJII, H., AND IWATA, M. Parallel processing of robot motion simulation. In *Proceedings IFAC World Congress, Munich* (1987).

[22] MALCZYK, P., AND FRĄCZEK, J. Lagrange multipliers based divide and conquer algorithm for dynamics of general multibody systems. In *Proceeding of the ECCOMAS Multibody Dynamics 2009 Thematic Conference, (CD-ROM)* (Warsaw, Poland, 29th June - 2nd July 2009), K. Arczewski, J. Frączek, and M. Wojtyra, Eds.

[23] MATTSON, T. G., SANDERS, B. A., AND MASSINGILL, B. L. *Patterns for Parallel Programming*. Addison-Wesley, Boston, 2005.

[24] MUKHERJEE, R., AND ANDERSON, K. Orthogonal complement based divide-and-conquer algorithm for constrained multibody systems. *Nonlinear Dynamics 48* (2007), 199–215.

[25] OPENMP. Application program interface, specification for parallel programming, version 3.0. Available online at http://www.openmp.org, May 2008.

[26] RODRIGUEZ, G. Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE Journal of Robotics and Automation 6* (1987), 624–639.

[27] WALKER, M. W., AND ORIN, D. E. Efficient dynamic computer simulation of robotic mechanisms. *ASME Journal of Dynamic Systems, Measurements, and Control 104* (1982), 205–211.