A R C H I V E   O F   M E C H A N I C A L   E N G I N E E R I N G

MARCIN PĘKAL [*], JANUSZ FRĄCZEK [*]

COMPARISON OF SELECTED FORMULATIONS FOR MULTIBODY SYSTEM DYNAMICS WITH REDUNDANT CONSTRAINTS

This paper compares selected optimization-based methods for the analysis of multibody systems with redundant constraints. The following numerical schemes are examined: direct integration method, Udwadia-Kalaba formulation, two types of least-squares block solution method and Udwadia-Phohomsiri formulation. In order to compare efficiency of the algorithms, a series of simulations is performed on two exemplary McPherson struts. In the first variant, the mechanism has no redundant constraints whereas the other is overconstrained. Three constraint stabilization schemes are also compared in terms of integration errors.

1. Introduction

Equations of motion for constrained multibody systems can be derived from various formalisms of mechanics, such as Lagrange, Hamilton or Appel's equations (Bauchau and Laulusa, 2008). As a result, in many of these formulations the equations of motions are written in the form of differential-algebraic equations (DAEs) (Amirouche, 2006; de Jalón and Bayo, 1994). If e.g. Lagrange's equations of the first kind are used and constraints are formulated on the position level then they can be considered as index-3 DAEs (Gear, 1988). In addition, it is often assumed that redundant constraints do not appear in the DAE system i.e. Jacobian matrix of the constraints equations has a full row rank. There are several classes in variety of numerical methods for DAE solution widely quoted in the literature, e.g. (Hairer and Wanner, 1996) originally formulated for such a case. Some of them rely on the direct application of numerical integration schemes (e.g. BDF or IRK methods) to the DAEs, in which the nonlinear algebraic equations are obtained and have

---
[*] *Warsaw University of Technology, Institute of Aeronautics and Applied Mechanics, Poland; E-mails: mpekal@meil.pw.edu.pl, jfraczek@meil.pw.edu.pl*

to be solved with Newton-like iterative methods. Other methods may rely on the transformation of DAEs into ordinary-differential equations (ODEs) which can be numerically integrated using classical ODE solvers. This transformation can be done e.g. through a suitable reduction to a minimal set of (independent) coordinates (de Jalón and Bayo, 1994).

However, if the system is overconstrained (and constraints are consistent), then a set of constraints equations is overdetermined, row rank of the Jacobian matrix of the constraints is not full and some or all Lagrange multipliers are not uniquely determined (Frączek and Wojtyra, 2011). As a consequence, many of the numerical methods mentioned above fail. There are a lot of proposals in literature on the subject to overcome this difficulty. One of the ideas, which are often used, is to determine acceleration and Lagrange multipliers vectors by means of optimization techniques which are insensitive to degeneration of the Jacobian matrix and nonuniqueness of the Lagrange multipliers. Methods for solving equations in the sense of least-squares are useful in this case, and particularly numerical techniques with direct application of pseudoinverse matrices or their suitable transformations (Eich-Soellner and Führer, 1998) are proposed.

In this paper, a comparison of selected numerical methods based on optimization idea is presented as applied to a DAE system featuring consistent redundant constraints. Numerical simulations are performed for two examples of spatial automotive suspensions. The first mechanism does not have redundant constraints, the second is overconstrained. Dependent-absolute (Cartesian) coordinates are used to represent kinematics of the systems. It results in mathematical models having the form of DAEs of index-1 (constraints on the acceleration level are taken). The following schemes are examined: direct integration method (de Jalón and Bayo, 1994), Udwadia-Kalaba formulation (de Falco *et al.*, 2009; Pennestrì and Valentini, 2007; Udwadia and Kalaba, 2001), two types of least-squares block solution method (de Falco *et al.*, 2009) and Udwadia-Phohomsiri formulation (de Falco *et al.*, 2009; Udwadia and Kalaba, 2001; Udwadia and Phohomsiri, 2006). For the mechanisms, three simulation cases are analyzed: without constraint stabilization (examined only to obtain the reference results), with the Baumgarte stabilization (Baumgarte, 1972), and with the coordinate partitioning method (Nikravesh, 1988). Moreover, all of the analyzes are carried out with various values of integration error tolerance (four values of integration errors are taken into consideration). In order to compare the efficiency of the mentioned methods, average values of computational time from 10 simulations are taken.

It is worth noting that similar studies have been already presented in literature, e.g. (Mariti *et.al.*, 2011, 2010; Pennestrì and Valentini, 2007; Pękal, 2012). The results obtained in this paper are compared with the outcomes

taken form two of them i.e. (Mariti *et.al.*, 2011, 2010). The first major difference with respect to the mentioned papers is associated with the fact that, in the citied papers, spatial systems are described using the set of four Euler parameters. In such a parametrization, the matrix $\mathbf{M}$ became singular, what did not allow one to obtain the results for several formulations like the Udwadia-Kalaba formulation and the least-squares 1 method there. Through the use of the three Euler angles for the rotation description of the spatial multibody systems, the methods which do not work due to the singularity of the inertia matrix $\mathbf{M}$ (the Udwadia-Kalaba formulation and the least-squares 1 method) are compared in this paper. In addition, three simulation cases are considered and direct integration method is analyzed. The paper presents in sequence: problem formulation, theoretical description of the considered methods, numerical examples, results of numerical tests and final conclusions.

2. Spatial multibody system dynamics

The general formulation of equations for multibody system dynamics in absolute coordinates can be found in many publications. In the present section, equations in the form analogous to e.g. (de Jalón and Bayo, 1994; Frączek and Wojtyra, 2008; Haug, 1989) are used. Moreover, it is assumed that the system may be overconstrained and constraints in the analytical form are consistent on all levels i.e. on the position, velocity and acceleration levels.

2.1. General equations of the dynamics

In the case of three-dimensional multibody system, absolute coordinates define the position and orientation of the body by means of the Cartesian coordinates of the origin of body-fixed local reference frame and by means of three Euler angles describing orientation of the local frame with respect to the inertial global reference frame. For constrained multibody systems, the additional equations – constraints are introduced in order to impose the motion requirements on the system. Using such description, the generalized coordinates vector for the multibody system can be written in the following form (Haug, 1989)

$$\mathbf{q}_{n\times 1} = [q_1 \ q_2 \ \dots \ q_n]^T = \left[\mathbf{q}_1^T \ \mathbf{q}_2^T \ \dots \ \mathbf{q}_N^T\right]^T, \tag{1}$$

where $\mathbf{q}_i = [x_i \ y_i \ z_i \ \alpha_i \ \beta_i \ \gamma_i]^T$, n is the number of generalized coordinates and N is the number of bodies in the examined system. Coordinates: x_i, y_i,

z_i describe i-th body translations (from the origin of the global reference frame to the origin of the i-th body local reference frame) and α_i, β_i, γ_i are aforementioned Euler angles (zxz): precession, nutation and intrinsic rotation angle respectively. They represent globally rotations of the i-th body.

The equation of motion can be written in the matrix form as (Udwadia and Kalaba, 2001)

$$\mathbf{M\ddot{q}} = \mathbf{Q} + \mathbf{W} \tag{2}$$

or (de Jalón and Bayo, 1994; Haug, 1989)

$$\mathbf{M\ddot{q}} + \mathbf{\Phi_q}^T\boldsymbol{\lambda} = \mathbf{Q}, \tag{3}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{Q}$ is the generalized forces vector (containing external and inertial terms), $\mathbf{W} = -\mathbf{\Phi_q}^T\boldsymbol{\lambda}$ is the generalized reactions vector, $\mathbf{\Phi_q}$ is the Jacobian matrix of the constraint equations described later and $\boldsymbol{\lambda}$ is the Lagrange multipliers vector.

The generalized coordinates need to fulfil the constraint equations at the position level (de Jalón and Bayo, 1994; Haug, 1989)

$$\mathbf{\Phi}_{m\times 1} = \mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0}, \tag{4}$$

where m is the number of constraints and t is time.
The foregoing constraint equations can be also expressed at the velocity level by differentiation of Eq. (4) with respect to time (Haug, 1989)

$$\dot{\mathbf{\Phi}} = \mathbf{\Phi_q}\dot{\mathbf{q}} + \mathbf{\Phi}_t = \mathbf{0}, \tag{5}$$

where the Jacobian matrix has the following form (de Jalón and Bayo, 1994)

$$\mathbf{\Phi_q}_{m\times n} = \frac{\partial\mathbf{\Phi}}{\partial\mathbf{q}} = \begin{bmatrix} \dfrac{\partial\Phi_1}{\partial q_1} & \dfrac{\partial\Phi_1}{\partial q_2} & \cdots & \dfrac{\partial\Phi_1}{\partial q_n} \\ \dfrac{\partial\Phi_2}{\partial q_1} & \dfrac{\partial\Phi_2}{\partial q_2} & \cdots & \dfrac{\partial\Phi_2}{\partial q_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial\Phi_m}{\partial q_1} & \dfrac{\partial\Phi_m}{\partial q_2} & \cdots & \dfrac{\partial\Phi_m}{\partial q_n} \end{bmatrix}_{m\times n}. \tag{6}$$

We denote the rank of the Jacobian matrix as r and let the Jacobian matrix be rank deficient.
Constraint equations can be also written at the acceleration level by differentiation of Eq. (5) with respect to time (Haug, 1989)

$$\ddot{\mathbf{\Phi}} = \mathbf{\Phi_q}\ddot{\mathbf{q}} - \mathbf{\Gamma} = \mathbf{0}, \tag{7}$$

where $\mathbf{\Gamma} = -(\mathbf{\Phi_q}\dot{\mathbf{q}})_\mathbf{q}\dot{\mathbf{q}} - 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}$.

The Eq. (3) together with the twice-differentiated constraint equations (Eq. (7)) constitute index-1 differential-algebraic equations (de Jalón and Bayo, 1994; Haug, 1989)

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{\Gamma} \end{bmatrix}. \tag{8}$$

Formula (8) is a linear equation with respect to the accelerations and Lagrange multipliers. It has a unique solution only when the row rank of the Jacobian matrix is full. This unique solution can be efficiently determined by exploiting sparsity of the coefficient matrix $\mathbf{X}$ (Torres-Moreno *et.al.*, 2013). The linear system (8) can be solved directly with respect to the vector of accelerations and Lagrange multipliers and the resultant accelerations can be next integrated as an ODE system. However, it is well known that this method of integration leads to the drift-off effect caused by the fact that Eqs (4) and (5) are not incorporated into error estimator of the ODE solver used.

Special methods are proposed in literature to improve accuracy of position and velocity constraints. In this paper, the attention is focused on two of them which are often used in computational practice during numerical integration of DAE. The Baumgarte stabilization method (Baumgarte, 1972), as one of the simplest and efficient algorithms is the first. When this method is used, the $\mathbf{\Gamma}$ vector is replaced by the modified expression (Haug, 1989)

$$\bar{\mathbf{\Gamma}} = \mathbf{\Gamma} - 2\hat{\alpha}\dot{\mathbf{\Phi}} - \hat{\beta}^2\mathbf{\Phi}, \tag{9}$$

where $\hat{\alpha}, \hat{\beta}$ are the Baumgarte stabilization parameters, which are chosen heuristically. These parameters are usually taken as $\hat{\alpha} = \hat{\beta} \in \langle 1, 20 \rangle$ (de Jalón and Bayo, 1994).

The second scheme of constraint stabilization is incorporated into the coordinate partitioning method that dates back to the paper (Wehage *et.al.*, 1981) used for DAE integration and implemented according to (Nikravesh, 1988). The coordinate partitioning is applied first. Redundant coordinates are partitioned into dependent (**u**) and independent (**v**) coordinates. Then only independent accelerations and velocities are integrated as ODEs. In each integration step, the dependent coordinates are computed using Newton-Raphson method based on the algorithm presented in e.g. (Haug, 1989) with a predetermined accuracy. This step provides stabilization of the position constraints violation error. Solving the linear equation (partitioned form of

Eq. (5) with respect to **u** and **v**) yields the dependent velocities. This stabilizes the velocity constraints. At the end of each integration step, the accelerations are calculated and integration may be continued.

When the examined system contains consistent redundant constraints, Jacobian matrix is rank deficient and values of the Lagrange multipliers (all or some) are no longer unique, what may affect not only the joint reactions but even the acceleration results for non-ideal systems e.g. when the friction is considered (Frączek and Wojtyra, 2011). In this case, some methods fail. The numerical methods analyzed in this paper operate also on rank deficient Jacobian matrices.

3. Numerical algorithms

Theoretical basis of the selected numerical methods, which allow constraint redundancy, are presented in this section.

3.1. Direct integration

The direct integration method uses directly index-1 DAEs (8) provided that the constraint equations are independent. During the integration process the equations containing unknown accelerations and Lagrange multipliers (included in the DAEs) are solved in each integration step, what may be expressed as (de Jalón and Bayo, 1994)

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\Gamma} \end{bmatrix}. \tag{10}$$

Note that due to effectiveness reasons the inverse matrix presented in the equation (10) is not calculated explicitly during simulations described below. Numerical calculations are carried out using solver for linear systems presented in (MATLAB®, help). The direct integration method is not applicable to overconstrained systems as for such systems the coordinate matrix **X** (See Eq. (8)) is singular and standard factorization algorithms fail. However, a method for overdetermined linear equations may be introduced to handle overconstrained systems. For example, the Moore-Penrose pseudoinverse matrix may be used. Thereby, the problem converts to optimization task which can be written in the form

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}^{+} \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\Gamma} \end{bmatrix}, \tag{11}$$

where $^+$ stands for Moore-Penrose (MP) pseudoinversion. It should be pointed out that MP pseudoinversion is used further in this paper. Note that in

many cases numerical cost of simulation is dominated by computation of the pseudoinverse solution.

3.2. Udwadia-Kalaba formulation

The Udwadia-Kalaba formulation uses a block inverse of the coefficient matrix appearing in Eq. (8) provided that there are no redundant constraints in the system and the matrix $\mathbf{M}$ is non-singular. This inversion, at a given claim, can be written in the form (Pennestrì and Valentini, 2007)

$$
\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} - \mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1} & \mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1} \\ (\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1} & -(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1} \end{bmatrix}.
\tag{12}
$$

Using expression (12), the solution of Eq. (8) takes the form

$$
\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{-1}\mathbf{Q} - \mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T[(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\mathbf{Q} - \boldsymbol{\Gamma})] \\ (\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\mathbf{Q} - \boldsymbol{\Gamma}) \end{bmatrix}.
\tag{13}
$$

Assume that (Udwadia and Kalaba, 2001)

$$
\ddot{\mathbf{q}}_f = \mathbf{M}^{-1}\mathbf{Q}
\tag{14}
$$

is the solution of the unconstrained system

$$
\mathbf{M}\ddot{\mathbf{q}}_f = \mathbf{Q}.
\tag{15}
$$

Substituting Eq. (14) into Eq. (13) leads to

$$
\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{q}}_f + \mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T(\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}(\boldsymbol{\Gamma} - \boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}}_f) \\ (\boldsymbol{\Phi}_{\mathbf{q}}\mathbf{M}^{-1}\boldsymbol{\Phi}_{\mathbf{q}}^T)^{-1}(\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}}_f - \boldsymbol{\Gamma}) \end{bmatrix}.
\tag{16}
$$

The next step is the inertia matrix $\mathbf{M}$ decomposition, what can be done using Cholesky (Banachiewicz) decomposition (Fortuna *et.al.*, 2005). Using the aforementioned decomposition, one can decompose an arbitrary symmetric, positive definite matrix $\mathbf{A}$ as the product of two matrices

$$
\mathbf{A} = \mathbf{L}\mathbf{L}^T,
\tag{17}
$$

where $\mathbf{L}$ is a lower triangular matrix. Taking $\mathbf{M}^{\frac{1}{2}} = \mathbf{L}$ matrix $\mathbf{M}$ can be written as

$$
\mathbf{M} = \mathbf{M}^{\frac{1}{2}}(\mathbf{M}^{\frac{1}{2}})^T.
\tag{18}
$$

(Note, that for the case of planar multibody systems the $\mathbf{M}$ matrix is diagonal and matrix $\mathbf{M}^{\frac{1}{2}}$ can be obtained directly).

The received $\mathbf{M}^{\frac{1}{2}}$ matrix is used to obtain the following matrix

$$\mathbf{M}^{-\frac{1}{2}} = \mathbf{M}^{-1}\mathbf{M}^{\frac{1}{2}}. \tag{19}$$

Let (Pennestrì and Valentini, 2007)

$$\mathbf{D} = \mathbf{\Phi_q}\mathbf{M}^{-\frac{1}{2}}. \tag{20}$$

Using Eqs (16) and (20) yields (Pennestrì and Valentini, 2007)

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_f + \mathbf{M}^{-\frac{1}{2}} \underbrace{\mathbf{M}^{-\frac{1}{2}}\mathbf{\Phi_q}^T}_{\mathbf{D}^T} (\underbrace{\mathbf{\Phi_q}\mathbf{M}^{-\frac{1}{2}}}_{\mathbf{D}} \underbrace{\mathbf{M}^{-\frac{1}{2}}\mathbf{\Phi_q}^T}_{\mathbf{D}^T})^{-1} (\mathbf{\Gamma} - \mathbf{\Phi_q}\ddot{\mathbf{q}}_f). \tag{21}$$

In order to simplify notation and due to the fact that inertia and Jacobian matrices are not rank deficient, the right inverse matrix definition

$$\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1} = \mathbf{D}^{\dagger} \tag{22}$$

can be noticed in the above formula, hence (Pennestrì and Valentini, 2007)

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_f + \mathbf{M}^{-\frac{1}{2}}\mathbf{D}^{\dagger}(\mathbf{\Gamma} - \mathbf{\Phi_q}\ddot{\mathbf{q}}_f). \tag{23}$$

Eventually final form of the equation can be written as

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{q}}_f + \mathbf{M}^{-\frac{1}{2}}\mathbf{D}^{\dagger}(\mathbf{\Gamma} - \mathbf{\Phi_q}\ddot{\mathbf{q}}_f) \\ (\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{\Phi_q}^T)^{-1}(\mathbf{\Phi_q}\ddot{\mathbf{q}}_f - \mathbf{\Gamma}) \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{q}}_f + \mathbf{M}^{-\frac{1}{2}}\mathbf{D}^{\dagger}(\mathbf{\Gamma} - \mathbf{\Phi_q}\ddot{\mathbf{q}}_f) \\ (\mathbf{D}\mathbf{D}^T)^{-1}(\mathbf{\Phi_q}\ddot{\mathbf{q}}_f - \mathbf{\Gamma}) \end{bmatrix}. \tag{24}$$

It should be emphasized that this formulation is valid only for systems containing non-singular matrix $\mathbf{M}$ (then $\mathbf{M}^{-1}$ exists) (Mariti *et.al.*, 2010) and Jacobian with full row rank (then the right inverse matrix $\mathbf{D}^{\dagger}$ and matrix $(\mathbf{D}\mathbf{D}^T)^{-1}$ exist). Therefore, for redundant multibody systems with non-singular $\mathbf{M}$ matrices, the formulation requires small modifications. Firstly, the MP pseudoinverse matrix will be used in the place of the right inverse appearing in Eq. (21) (which can be used for non–overconstrained systems). Secondly, the inversion $(\mathbf{D}\mathbf{D}^T)^{-1}$ in Eq. (24) should be replaced consequently by MP pseudoinverse $(\mathbf{D}\mathbf{D}^T)^{+}$.

3.3. Least-squares block solution

Assume that the matrix $\mathbf{M}$ has full rank. Hence, it is possible to create the following pseudoinverse matrix as (de Falco *et al.*, 2009)

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix}^{+} = \begin{bmatrix} \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{\Phi_q}^T(\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{\Phi_q}^T)^{+}\mathbf{\Phi_q}\mathbf{M}^{-1} & \mathbf{M}^{-1}\mathbf{\Phi_q}^T(\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{\Phi_q}^T)^{+} \\ (\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{\Phi_q}^T)^{+}\mathbf{\Phi_q}\mathbf{M}^{-1} & -(\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{\Phi_q}^T)^{+} \end{bmatrix}. \tag{25}$$

Note that this expression is analogous to Eq. (12).

The following assumption (de Falco *et al.*, 2009)

$$\mathbf{Q}_{LS1} = \mathbf{\Phi_q} \mathbf{M}^{-1} \mathbf{\Phi_q}^T \qquad (26)$$

yields (de Falco *et al.*, 2009)

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix}^+ = \begin{bmatrix} \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{\Phi_q}^T\mathbf{Q}_{LS1}^+\mathbf{\Phi_q}\mathbf{M}^{-1} & \mathbf{M}^{-1}\mathbf{\Phi_q}^T\mathbf{Q}_{LS1}^+ \\ \mathbf{Q}_{LS1}^+\mathbf{\Phi_q}\mathbf{M}^{-1} & -\mathbf{Q}_{LS1}^+ \end{bmatrix}. \qquad (27)$$

Hence, the solution of Eq. (8) can be written as

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix}^+ \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\Gamma} \end{bmatrix}, \qquad (28)$$

which, after applying Eq. (27), gives

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} (\mathbf{I} - \mathbf{M}^{-1}\mathbf{\Phi_q}^T\mathbf{Q}_{LS1}^+\mathbf{\Phi_q})\mathbf{M}^{-1}\mathbf{Q} + \mathbf{M}^{-1}\mathbf{\Phi_q}^T\mathbf{Q}_{LS1}^+\boldsymbol{\Gamma} \\ \mathbf{Q}_{LS1}^+(\mathbf{\Phi_q}\mathbf{M}^{-1}\mathbf{Q} - \boldsymbol{\Gamma}) \end{bmatrix}, \qquad (29)$$

where $\mathbf{I}$ is the unit matrix. This form of the least-squares block solution will be denoted as the least-squares 1 method.

It is necessary to add that the least-squares 1 method is not valid for equations with matrix $\mathbf{M}$ being singular. In order to overcome this problem, the second type of least-squares block solution method is used. This formulation uses directly the fact that mass matrix is symmetric.

Assume that (de Falco *et al.*, 2009)

$$\mathbf{R} = \mathbf{M}\mathbf{\Phi_q}^+, \qquad (30)$$

$$\mathbf{E} = \mathbf{I} - \mathbf{\Phi_q}^+\mathbf{\Phi_q}. \qquad (31)$$

Note that when the $\mathbf{\Phi_q}$ is a full–rank matrix, then $\mathbf{\Phi_q}^+\mathbf{\Phi_q} = \mathbf{I}$ and, as result, $\mathbf{E} = \mathbf{0}$.

Let also

$$\mathbf{Q}_{LS2} = (\mathbf{EME})^+. \qquad (32)$$

Then (de Falco *et al.*, 2009)

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix}^+ = \begin{bmatrix} \mathbf{0} & \mathbf{\Phi_q}^+ \\ (\mathbf{\Phi_q}^+)^T & -(\mathbf{\Phi_q}^+)^T\mathbf{R} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ -\mathbf{R}^T \end{bmatrix} \mathbf{Q}_{LS2}[\mathbf{I} \quad -\mathbf{R}]. \qquad (33)$$

Hence, Eq. (28) can be written as

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{LS2}\mathbf{Q} + (\mathbf{\Phi_q}^+ - \mathbf{Q}_{LS2}\mathbf{R})\boldsymbol{\Gamma} \\ [(\mathbf{\Phi_q}^+)^T - \mathbf{R}^T\mathbf{Q}_{LS2}]\mathbf{Q} + [-(\mathbf{\Phi_q}^+)^T + \mathbf{R}^T\mathbf{Q}_{LS2}]\mathbf{R}\boldsymbol{\Gamma} \end{bmatrix}. \qquad (34)$$

This scheme will be denoted as the least-squares 2 method.

It is worth noting that both types of the least-squares block solution method are applicable to redundant systems analysis.

3.4. Udwadia-Phohomsiri formulation

The Udwadia-Phohomsiri formulation was originally proposed for the system with non-ideal constraints and for the matrix equation of motion in the form of (2). The original form of these equations is maintained here despite the fact that, in this paper, the systems with ideal constraints are analyzed only.

According to d'Alembert's principle for the ideal constraints, the work done in virtual displacements is equal zero. However, when there are non-ideal constraints in the system, the situation is more complex (Udwadia and Phohomsiri, 2006)

$$(\delta \mathbf{w})^T \mathbf{W} = (\delta \mathbf{w})^T \mathbf{C}, \tag{35}$$

where $\mathbf{C}$ is the vector which describes the nature of the non-ideal constraints (for the ideal constraints $\mathbf{C} \equiv \mathbf{0}$) and $\delta \mathbf{w}$ is the virtual displacement vector. When constraints have the form (4), the virtual displacement vector fulfils the following expression (Udwadia and Phohomsiri, 2006)

$$\mathbf{\Phi_q} \delta \mathbf{w} = \mathbf{0}, \tag{36}$$

for which the solution is (See Eq. (31)) (Udwadia and Phohomsiri, 2006)

$$\delta \mathbf{w} = (\mathbf{I} - \mathbf{\Phi_q^+} \mathbf{\Phi_q}) \delta \boldsymbol{\gamma} = \mathbf{E} \delta \boldsymbol{\gamma}, \tag{37}$$

where $\delta \boldsymbol{\gamma}$ is arbitrary.

Note that the above equation contains only the non-unique part of the general solution of the linear system. This part results from the nullspace of the Jacobian matrix $\mathbf{\Phi_q}$.

Substituting Eq. (37) into Eq. (35) and using $\mathbf{E}^T = \mathbf{E}$ yields (Udwadia and Phohomsiri, 2006)

$$(\delta \boldsymbol{\gamma})^T \mathbf{E} \mathbf{W} = (\delta \boldsymbol{\gamma})^T \mathbf{E} \mathbf{C}. \tag{38}$$

Note that $\delta \boldsymbol{\gamma}$ is independent, therefore the above equation can be simplified to the form (Udwadia and Phohomsiri, 2006)

$$\mathbf{E} \mathbf{W} = \mathbf{E} \mathbf{C}. \tag{39}$$

Left-side multiplying of Eq. (2) by $\mathbf{E}$ and using Eq. (39) yields the formula which can be combined with Eq. (7) as (Udwadia and Phohomsiri, 2006)

$$\begin{bmatrix} \mathbf{E} \mathbf{M} \\ \mathbf{\Phi_q} \end{bmatrix} \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{E}(\mathbf{Q} + \mathbf{C}) \\ \mathbf{\Gamma} \end{bmatrix}. \tag{40}$$

Denote (Udwadia and Phohomsiri, 2006)

$$\bar{\mathbf{M}}_{(n+m)\times n} = \begin{bmatrix} \mathbf{EM} \\ \mathbf{\Phi_q} \end{bmatrix}. \tag{41}$$

Solving Eq. (40) yields (Udwadia and Phohomsiri, 2006)

$$\ddot{\mathbf{q}}_{n\times 1} = \bar{\mathbf{M}}^+_{n\times(n+m)} \begin{bmatrix} \mathbf{Q} + \mathbf{C} \\ \mathbf{\Gamma} \end{bmatrix}_{(n+m)\times 1} + (\mathbf{I}_{n\times n} - \bar{\mathbf{M}}^+_{n\times(n+m)}\bar{\mathbf{M}}_{(n+m)\times n})\boldsymbol{\eta}_{n\times 1}, \tag{42}$$

where $\boldsymbol{\eta}$ is arbitrary.

The above statement has infinitely many solutions, in general. But, when $\bar{\mathbf{M}}$ has full rank, then $\bar{\mathbf{M}}^+\bar{\mathbf{M}} = \mathbf{I}$, because the nullspace of $\bar{\mathbf{M}}$ is empty. Hence, Eq. (42) becomes uniquely solvable and the solution takes the form (Udwadia and Phohomsiri, 2006)

$$\ddot{\mathbf{q}} = \bar{\mathbf{M}}^+ \begin{bmatrix} \mathbf{Q} + \mathbf{C} \\ \mathbf{\Gamma} \end{bmatrix}. \tag{43}$$

Note that this method is also suitable for redundant multibody systems. Moreover, it is useful to add that, when ideal systems are considered, the acceleration vector is always unique.

4. Numerical examples

4.1. Selected mechanisms

In order to compare the effectiveness of the considered methods, two automotive suspension mechanisms are examined. The simplified McPherson strut shown in Fig. 1 is the first one and the overconstrained McPherson strut presented in Fig. 2 is the second. This type of the McPherson strut is taken from (Haug, 1989). The first mechanism is a variant of the second system obtained by the elimination of the redundant constraints, i.e. by the elimination of one of the revolute joints between the ground and the first body. Absolute coordinates with the set of 3 Euler angles (zxz convention) are used in the kinematic description of these mechanisms. Moreover, the formulation with such coordinates and their derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ leads to non-singular matrix $\mathbf{M}$ is used.

Both mechanisms consist of 5 rigid bodies and they have 4 degrees of freedom (DOFs). Note that 2 of the DOFs are local mobilities of bodies 3 and 4. The characteristic points are depicted in Figs 1 and 2. The main dimensions of the systems are: $|P_1P_2| = 0.3$ m, $|P_1P_{1A}| = 0.025$ m, $|P_1P_{1B}| = 0.025$ m, $|P_3P_4| = 0.45$ m, $|P_3P_5| = 0.1$ m, $|P_3P_6| = 0.25$ m, $|P_2P_3| = 0.05$ m, $|P_6P_7| = 0.1$ m, $|P_5P_8| = 0.15$ m, $|P_7P_9| = 0.05$ m, $|P_{10}P_{11}| =$
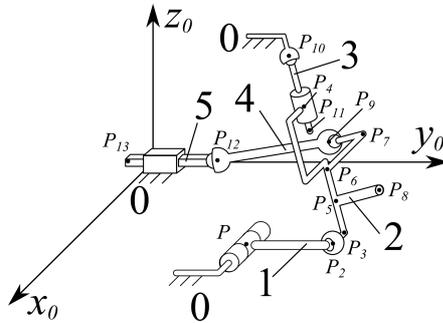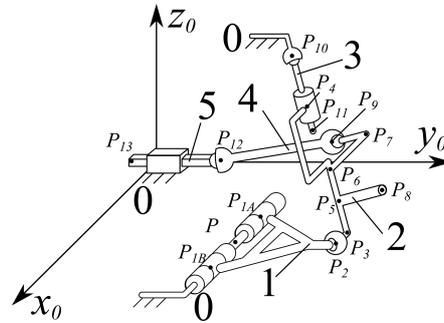
Fig. 1. Simplified McPherson strut

Fig. 2. Overconstrained McPherson strut

$= 0.3$ m, $|P_9P_{12}| = 0.5$ m, a $|P_{12}P_{13}| = 0.2$ m. In order to simplify the description of the mechanisms, local coordinate systems are placed in the bodies' centres of masses. Moreover, the centres of masses for bodies 1-5 are located in the middle of the line segments: P_1P_2, P_3P_4, $P_{10}P_{11}$, P_9P_{12} and $P_{12}P_{13}$ respectively. It is assumed that all bodies have equal masses $m = 1$ kg and non-zero moments of inertia: $J_{xx} = 0.1$ kgm^2, $J_{yy} = 0.1$ kgm^2 and $J_{zz} = 0.1$ kgm^2.

The mechanisms are loaded in the same manner, i.e., by means of $F_z = = 38$ N force applied to the centre of mass of the body 1 and $F_y = 2 + 0.01 \cdot \cdot \sin(t)$ N force applied to the centre of mass of the body 5. The forces act in the z and y directions of the global reference system respectively. Gravity force is considered in the simulations. It acts in the z direction of the global coordinate system with acceleration $g = -9.80665$ m/s^2.

All examined algorithms were implemented in the MATLAB$^{®}$ R2014b package. Differential equations were integrated by *ode*45 function which is a variant of the Runge-Kutta algorithm. Four values of the integration error tolerance were considered: *AbsTol* = *RelTol* $\in$ {1.00e-3, 1.00e-6, 1.00e-8, 1.00e-10} (*AbsTol* is the absolute error tolerance and *RelTol* is the relative error tolerance (MATLAB$^{®}$, help)). The simulations for the time of 10 s motion were performed. These simulations can be divided into three groups: without stabilization of the constraints, with the Baumgarte stabilization method and with the coordinate partitioning method. It was taken $\hat{\alpha} = \hat{\beta} = 10$ for the Baumgarte stabilization. For the coordinate partitioning method, which uses the Newton-Raphson scheme, tolerance of the norm of the position constraints (*NormTol*) for the Newton-Raphson algorithm was three orders of magnitude smaller than the integrator tolerance. For simplicity, coordinates were partitioned only once, at the beginning of each analysis. In order to assess the correctness of the partition, the conditioning of the submatrix of the Jacobian matrix (which corresponds to the dependent coordinates) was monitored during the simulation. Moreover, because of rank

deficiency of the Jacobian matrix, its inversion occurring in the correction phase of the Newton-Raphson method, was replaced with MP pseudoinverse.

MATLAB® R2014b was run in the single thread mode in order to avoid time measurement problems. The computational time was measured by *tic* and *toc* functions. Ten simulations (integrations) were performed and the average computation time was determined in each case.

The correctness of the obtained results was verified by comparing them to results of independent dynamics simulations in the 64–bit version of the Adams™ 2014.0.1 program with appropriately selected integration error tolerances.

All the computations were performed using the computer equipped with Intel® CORE™ i5 CPU M520 @ 2.40 GHz 2.40 GHz processor, 4 GB of RAM and 64-bit Microsoft® Windows® 7 Home Premium operating system.

4.2. Results

Absolute average computation time and number of right-hand side (RHS) computations (carried out by the integrator) from 10 simulation runs (for each case) are shown in the following figures. The outcomes from simulations performed without constraint stabilization are presented in Figs 3 and 4. The Figs 5 and 6 show the results obtained using the Baumgarte stabilization method, while Figs 7 and 8 depict outcomes received by applying the coordinate partitioning method. Moreover, the exact numerical values of the computation time from these figures are also presented in the Table 1, which is placed in appendix A.
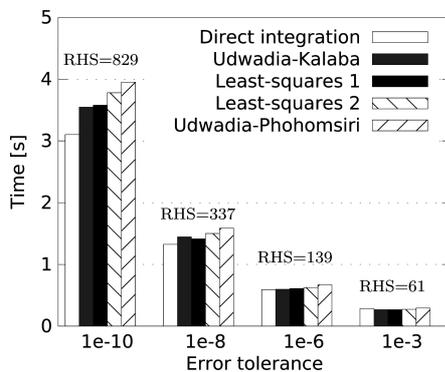

Fig. 3. Time and number of right-hand side computations of simulations for simplified McPherson strut without constraint stabilization
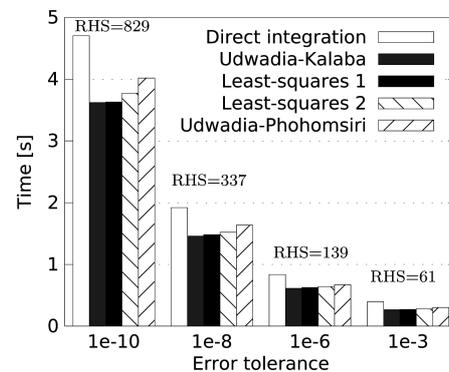
Fig. 4. Time and number of right-hand side computations of simulations for overconstrained McPherson strut without constraint stabilization
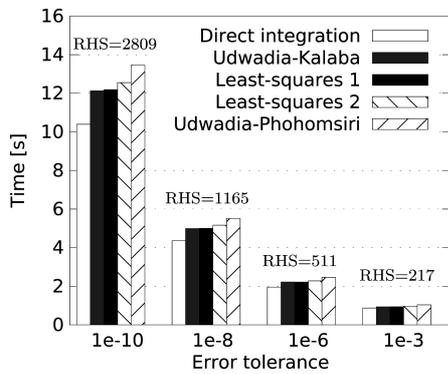
Fig. 5. Time and number of right-hand side computations of simulations for simplified McPherson strut with the Baumgarte stabilization
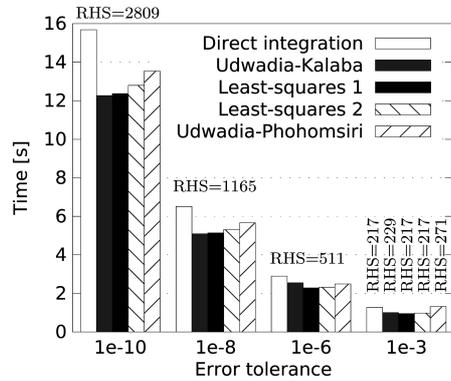

Fig. 6. Time and number of right-hand side computations of simulations for overconstrained McPherson strut with the Baumgarte stabilization

The errors in constraints are expressed via Euclidean norms, as shown in Figs 9, 10 and 11. These figures contain the constraint norms at the position, velocity and acceleration level for all stabilization cases obtained from simplified McPherson strut analysis by the least-squares 2 method for the integration error tolerance *AbsTol* = *RelTol* = 1.00e-10. The Fig. 9 presents results of simulation without constraint stabilization. The Figs 10 and 11 show the results of simulations when the Baumgarte and the coordinate partitioning methods of constraint stabilization are applied respectively.
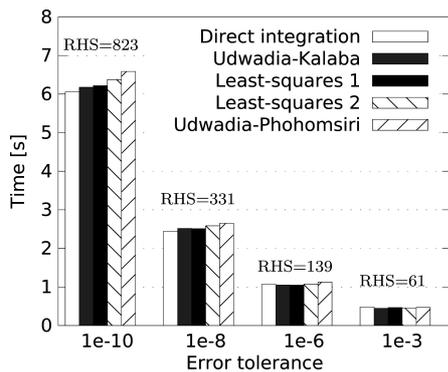

Fig. 7. Time and number of right-hand side computations of simulations for simplified McPherson strut with the coordinate partitioning method
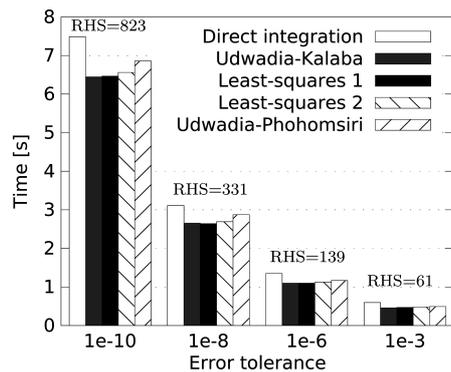

Fig. 8. Time and number of right-hand side computations of simulations for overconstrained McPherson strut with the coordinate partitioning method

The computational time of simulation without constraint stabilization is shorter than the corresponding time for analyses with stabilization. The accuracy of the position and velocity constraints degrades quickly for the unstabilized simulations, what can be observed in Fig. 9.

Computational time for analyses with the coordinate partitioning method is about half as long as simulations without stabilization and, in turn, twice shorter than time of simulations with the Baumgarte stabilization. Note that the above observation is valid provided that the coordinate partitioning is made only once, at the beginning of each simulation. Additionally, it should be added that Newton-Raphson method appearing in the coordinate partitioning algorithm needs typically 3 or 4 iterations to achieve the assumed accuracy of the position constraints. It should be also emphasized that the norms for both stabilization methods do not reveal the drift-off effect, what can be seen in Figs 10 and 11.
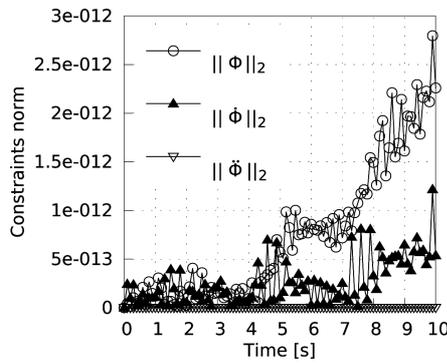

Fig. 9. Constraint norms for simplified McPherson strut without constraint stabilization for the selected case
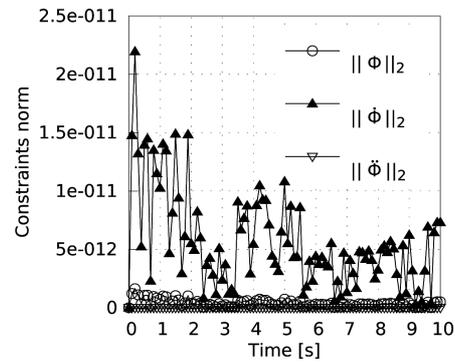

Fig. 10. Constraint norms for simplified McPherson strut with the Baumgarte stabilization for the selected case
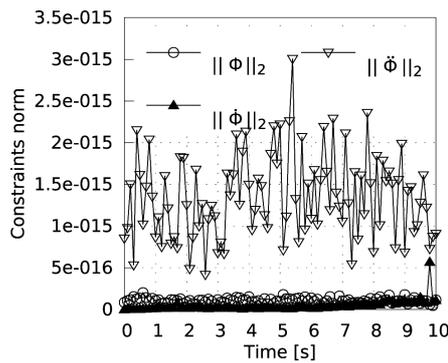

Fig. 11. Constraint norms for simplified McPherson strut with the coordinate partitioning method for the selected case

According to the simulation results, the direct integration method turns out to be the fastest for simplified McPherson strut. The Udwadia-Kalaba formulation and the least-squares 1 method are slightly less effective than the direct integration method there. Additionally, the least-squares 2 method

gives slightly worse timing results. Moreover, the Udwadia-Phohomsiri formulation gives the slowest timing results for this mechanism. For the system with redundant constraints, the direct integration method is the least efficient in most cases whereas the Udwadia-Kalaba formulation and the least–squares 1 methods are the most effective (with slight differences).

As mentioned above, the number of RHS computations of simulations is also presented in Figs. 3–8. Note that the RHS computations number is the same for all considered methods in each computation case in general. There is one exception. For overconstrained McPherson Strut with the Baumgarte stabilization and error tolerance equal to 1e-3 there are differences in the number of RHS computations between methods (See Fig. 6). It may be due to numerical errors at a high error tolerance equal to 1e-3. The RHS data may be used in order to show time relative to number of RHS computations. Conducting this does not change the above observations about efficiency even for the exceptional case.

Some of the results confirm suggestions formulated in the literature. Most of the considered numerical methods were analyzed in (Mariti *et.al.*, 2011, 2010), where Euler parameters were used to express angular orientation (in the case of spatial multibody systems). In the formulation presented there, matrix **M** was singular, therefore the Udwadia-Kalaba formulation and the least–squares 1 could not be applied to spatial mechanisms. In the mentioned publications, the least-squares 2 method was faster than the Udwadia-Phohomsiri formulation for the spatial system without redundant constraints. This agrees with the results obtained in this paper.

5. Conclusions

Comparison of the selected methods for constrained multibody dynamics with redundant constraints has been presented. The following methods have been examined: direct integration method, Udwadia-Kalaba formulation, two types of the least-squares block solution method and Udwadia-Phohomsiri formulation. Two exemplary rigid body mechanisms: simplified McPherson strut and overconstrained McPherson strut have been simulated. Three simulation sets have been considered: without constraint stabilization, with the Baumgarte stabilization and with the coordinate partitioning method. All analyses have been conducted for four different integration error tolerances. The obtained results have been confronted with suggestions formulated in the literature (Mariti *et.al.*, 2011, 2010).

Three methods: the Udwadia-Kalaba and both least-squares, have had similar effectiveness measured by computational time in all the analyses. Direct integration method (which seems to be the simplest) has been the fastest in most cases for the simplified McPherson strut, whereas, for overconstrained

system, this method has been least effective. Therefore, it is useful to take it into consideration when analyzing non-overconstrained systems. Time of simulation for the Udwadia-Kalaba formulation and the least-squares 1 method have been slightly shorter than in other methods for the overconstrained system. It should be pointed out that for systems with singular matrix **M**, these two methods cannot be applied. However, such systems can be analyzed using the least-squares 2 method. Moreover, the Udwadia-Phohomsiri formulation have had relatively weak effectiveness for both mechanisms. It seems that this method is not a rival for systems containing ideal constraints only, but it is appropriate for systems containing non-ideal constraints (according to its authors) e.g. when there is friction in a mechanism. Note that such systems are not considered in this paper.

As expected, the times of simulations with constraint stabilization were longer than times of analyses without stabilization. In the case of the simulation with constraint stabilization, additional computations have to be performed, increasing the computational cost. On the other hand, the accuracy of the position and velocity constraints decreases rapidly for the simulations without stabilization. Hence, this approach is not suitable for practical applications (where the longer time is considered), however, it is used to provide reference performance measures. In addition, simulations with the Baumgarte stabilization turned out to be slower than analyses with the coordinate partitioning. This conclusion is not consistent with the experience of other authors. This outcome could results from the fact that the coordinate partitioning has been performed once per simulation. Hence, the coordinate partitioning updating process, which is numerically expensive, has been avoided. In practical applications, covering large multibody systems, the coordinate partitioning is carried out automatically and is updated during the simulation in general. This can significantly reduce the efficiency of this stabilization algorithm. Furthermore, in all calculations libraries dedicated to sparse matrices have not been used. For the case of large multibody systems, sparse techniques can significantly decrease computational cost.

The presented conclusions are true for the two exemplary mechanisms. Therefore, they shall not be directly generalized into the case of more complex, arbitrary multibody systems. However, the results may be useful for method selection for solving the equations of motion of mechanisms without or with redundant constraints.

A Appendix

Table 1.

Exact numerical results presented in the figures

Reference Figure	Error Tolerance	Computational Time				
		Direct integration	Udwadia-Kalaba	Least-squares 1	Least-squares 2	Udwadia-Phohomsiri
Figure 3	1e-10	3.10	3.55	3.58	3.78	3.95
	1e-8	1.33	1.45	1.41	1.50	1.59
	1e-6	0.59	0.60	0.61	0.63	0.67
	1e-3	0.29	0.27	0.27	0.27	0.29
Figure 4	1e-10	4.70	3.62	3.63	3.77	4.02
	1e-8	1.92	1.46	1.48	1.52	1.64
	1e-6	0.84	0.61	0.63	0.64	0.67
	1e-3	0.40	0.27	0.27	0.28	0.30
Figure 5	1e-10	10.39	12.13	12.19	12.56	13.46
	1e-8	4.35	5.00	5.01	5.17	5.52
	1e-6	1.96	2.22	2.22	2.28	2.46
	1e-3	0.88	0.94	0.94	0.97	1.04
Figure 6	1e-10	15.69	12.27	12.37	12.80	13.54
	1e-8	6.51	5.09	5.14	5.31	5.67
	1e-6	2.90	2.55	2.29	2.32	2.49
	1e-3	1.27	1.00	0.95	0.98	1.33
Figure 7	1e-10	6.05	6.18	6.22	6.38	6.59
	1e-8	2.45	2.52	2.51	2.59	2.65
	1e-6	1.07	1.05	1.05	1.07	1.12
	1e-3	0.48	0.45	0.47	0.46	0.47
Figure 8	1e-10	7.49	6.45	6.47	6.56	6.86
	1e-8	3.11	2.66	2.64	2.69	2.87
	1e-6	1.35	1.10	1.10	1.13	1.17
	1e-3	0.60	0.46	0.47	0.48	0.49

REFERENCES

[1] Amirouche F.M.L.: Fundamentals of Multibody Dynamics, Theory and Applications, Birkhäuser Boston, 2006.

[2] Bauchau O.A., Laulusa A.: Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems, Journal of Computational and Nonlinear Dynamics, 3, 1, 011005 (1-8), 2008.

[3] Baumgarte J.W.: Stabilization of constraints and integrals of motion in dynamical systems, Computer Methods in Applied Mechanics and Engineering, 1, 1, 1972, 1-16.

[4] de Falco D., Pennestrì E., Vita L.: Investigation of the Influence of Pseudoinverse Matrix Calculations on Multibody Dynamics Simulations by Means of the Udwadia-Kalaba Formulation, Journal of Aerospace Engineering, 22, 4, 2009, 365-372.

[5] de Jalón J.G., Bayo E.: Kinematic and Dynamic Simulation of Multibody Systems, The Real-Time Challenge, Springer-Verlag, New-York, 1994.

[6] Eich-Soellner E., Führer C.: Numerical Methods in Multibody Dynamics, B.G. Teubner, 1998.

[7] Fortuna Z., Macukow B., Wąsowski J.: Metody numeryczne (Numerical methods). Wydawnictwa Naukowo-Techniczne, Warszawa, 2005 (in Polish).

[8] Frączek J., Wojtyra M.: Kinematyka układów wieloczłonowych. Metody obliczeniowe (Multibody kinematics. Computational methods), Wydawnictwa Naukowo-Techniczne, Warszawa, 2008 (in Polish).

[9] Frączek J., Wojtyra M.: On the unique solvability of a direct dynamics problem for mechanisms with redundant constraints and Coulomb friction in joints, Mechanism and Machine Theory, 46, 3, 2011, 312-334.

[10] Gear C.W.: Differential-Algebraic Equation Index Transformations, SIAM Journal on Scientific and Statistical Computing, 9, 1, 1988, 39-47.

[11] Hairer E., Wanner G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer Berlin Heidelberg, 2 edition, 1996.

[12] Haug E.J.: Computer aided kinematics and dynamics of mechanical systems, Volume I: Basic methods, Allyn and Bacon, 1989.

[13] Mariti L., Belfiore N.P., Pennestrì E., Valentini P.P.: Comparison of solution strategies for multibody dynamics equations, International Journal for Numerical Methods in Engineering, 88, 7, 2011, 637-656.

[14] Mariti L., Pennestrì E., Valentini P.P., Belfiore N.P.: Review and Comparison of Solution Strategies for Multibody Dynamics Equations, The 1st Joint International Conference on Multibody System Dynamics, Lappeenranta, Finland, 2010.

[15] MATLAB®, help.

[16] Nikravesh P.E.: Computer-Aided Analysis of Mechanical Systems, Prentice Hall, 1988.

[17] Pennestrì E., Valentini P.P.: Coordinate reduction strategies in multibody dynamics: A review, Proceedings of the Conference on Multibody System Dynamics, 2007.

[18] Pękal M.: Porównanie metod całkowania RRA w analizie dynamiki układów wieloczłonowych (Comparison of DAE integration methods for multibody dynamics). MSc Thesiss. Praca magisterska. Politechnika Warszawska, Warszawa, 2012 (in Polish).

[19] Torres-Moreno J.L., Blanco J.L., López-Martínez J., Giménez-Fernández A.: A comparison of Algorithms for Sparse Matrix Factoring and Variable Reordering aimed at Real-time Multibody Dynamic Simulation, ECCOMAS Multibody Dynamics, ECCOMAS, 2013, 167-168.

[20] Udwadia F.E., Kalaba R.E.: Explicit Equations of Motion for Mechanical Systems With Nonideal Constraints, ASME Journal of Applied Mechanics, 68, 3, 2001, 462-467.

[21] Udwadia F.E., Phohomsiri P.: Explicit equations of motion for constrained mechanical systems with singular mass matrices and applications to multi-body dynamics, Proceedings of the Royal Society A, 462, 2071, 2006, 2097-2117.

[22] Wehage R.A., Haug E.J., Beck R.R.: Generalized Coordinate Partitioning in Dynamic Analysis of Mechanical Systems, Technical Report 12590, The University of Iowa, 1981.

Porównanie wybranych metod analizy dynamiki układów wieloczłonowych z więzami nadmiarowymi

S t r e s z c z e n i e

W pracy porównano wybrane metody analizy układów wieloczłonowych z więzami nadmiarowymi z zastosowaniem metod optymalizacji. Porównano następujące metody całkowania równań ruchu: metodę bezpośredniego całkowania, dwa rodzaje metody najmniejszych kwadratów oraz trzy różne sformułowania przedstawione w piśmiennictwie. Przeanalizowano trzy przypadki stabilizacji rozwiązania, przeprowadzono serie symulacji na dwóch przykładach mechanizmów kolumny McPhersona. Sformułowano wnioski co do efektywności rozwiązania układów równań ruchu mechanizmów, które mogą być przydatne dla doboru odpowiedniej metody całkowania.