

COMPARISON OF NATURAL COMPLEMENT FORMULATIONS FOR MULTIBODY DYNAMICS

MARCIN PEKAL, JANUSZ FRĄCZEK

Warsaw University of Technology, Institute of Aeronautics and Applied Mechanics, Poland
e-mail: mpekal@meil.pw.edu.pl; jfraczek@meil.pw.edu.pl

The main aim of this paper is to compare the effectiveness of numerical techniques used for spatial multibody dynamics simulation by applying the natural complement method. In the paper, seven numerical schemes are considered: zero eigenvalue formulation, Pseudo Upper Triangular Decomposition, Schur decomposition, Singular Value Decomposition, QR decomposition, coordinate partitioning and Wang-Huston formulation. In order to illustrate the effectiveness of these schemes, two McPherson struts are considered. Simulations are performed with four error tolerance values and for three stabilization cases. Some suggestions on possible applications of the selected methods are formulated.

Keywords: multibody dynamics, numerical methods, spatial systems

1. Introduction

The equations of motion of multibody systems may be written using many different coordinate sets (de Jalón and Bayo, 1994; Malczyk and Frączek, 2012). However, the most frequently used are absolute, natural and relative coordinates. Absolute and natural coordinates constitute redundant sets of coordinates hence a system of equations of motion with constraints has to be considered. Such a system is the set of differential-algebraic equations (DAEs) with differential index at most equal to 3. Numerical methods used to solve DAEs are still under intensive development and the effective integration procedures are less well-known for them than for ordinary differential equations (ODEs) (Kunkel and Mehrman, 2006). Note that there exist publications about differential equations in mechanical systems, e.g. (Awrejcewicz, 2014; Eichsoellner and Führer, 1998).

Many integration methods can be used to solve DAEs (de Jalón and Bayo, 1994; Haug, 1989). A large group of these methods transforms the equations of motion described in redundant coordinates into equations written in the minimal set of coordinates. To perform this transformation, the null space base of the constraints manifold is built and afterwards, equations of motion are expressed in this base. Null space base vectors are represented by the orthogonal complementary matrix. Finally, the set of ODEs is obtained, which can be solved efficiently. There are several methods proposed in literature to accomplish the orthogonal complement.

The main objective of this paper is to compare numerical schemes for DAEs based on or equivalent to the orthogonal complement methods using two examples. The following methods are considered: zero eigenvalue formulation, Pseudo Upper Triangular Decomposition (PUTD), Schur decomposition, Singular Value Decomposition (SVD), QR decomposition, coordinate partitioning and Wang-Huston formulation. Hereafter, spatial rigid multibody systems described with the absolute coordinates are considered. The Euler angles (consecutive rotations about the z , x and z axes) are used for the orientation description and consequently, the mass matrix is nonsingular. For the chosen coordinates set, the resulting equations of motion are in the form of differential-algebraic equations that consist of differential equations of motion and algebraic

equations of constraints. Note that there are also papers analyzing impact, e.g. (Awrejcewicz *et al.*, 2003, 2004; Awrejcewicz and Kudra, 2005) or contact, e.g. (Awrejcewicz and Kudra, 2014), but these phenomena are not examined here.

In order to compare the efficiency of the examined methods, two types of McPherson struts are analyzed. The first model of the strut has no redundant constraints, while in the second mechanism the redundancy is taken into consideration (Wojtyra and Frączek, 2013). For all the considered methods, simulations are performed with four values of the error tolerances and for three constraint stabilization approaches (with no stabilization of the constraints, with the Baumgarte stabilization and with the position constraints stabilization using the Newton-Raphson method). It is worth noting that similar comparisons were already performed by other authors (Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007) as well as in (Pełal, 2012). Therefore, the comparison with the previous publications (Mariti *et al.*, 2011, 2010) is also performed.

2. Spatial system dynamics

Multibody dynamics description presented in this Section is based on the absolute coordinate formulation and appears in, e.g. (Frączek and Wojtyra, 2008; de Jalón and Bayo, 1994; Haug, 1989).

The vector of the absolute generalized coordinates can be written as (Haug, 1989): $\mathbf{q}_{n \times 1} = [\mathbf{q}_1^T \ \mathbf{q}_2^T \ \cdots \ \mathbf{q}_N^T]^T = [q_1 \ q_2 \ \cdots \ q_n]^T$, where N is the number of bodies and n is the number of generalized coordinates of the multibody system. Moreover, the equation of motion has the following form (de Jalón and Bayo, 1994; Haug, 1989)

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q}^e \quad (2.1)$$

where \mathbf{M} is the mass matrix, $\Phi_{\mathbf{q}}$ is the Jacobian matrix of constraints (see Eq. (2.3)), $\boldsymbol{\lambda}$ is the Lagrange multipliers vector and \mathbf{Q}^e is the generalized force vector. Matrix equation (2.1) is the system of n equations with $n + m$ unknowns ($\ddot{\mathbf{q}}_{n \times 1}$ and $\boldsymbol{\lambda}_{m \times 1}$), where m is the number of Lagrange multipliers. To solve this system, the additional m constraint equations should be introduced as (de Jalón and Bayo, 1994; Haug, 1989)

$$\Phi_{m \times 1} = \Phi(\mathbf{q}, t) = \mathbf{0} \quad (2.2)$$

where t denotes time. After differentiating (2.2) over time, we obtain (Haug, 1989)

$$\dot{\Phi} = \Phi_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t = \mathbf{0} \quad (2.3)$$

Differentiation of Eq. (2.3) once again over time yields (Haug, 1989)

$$\ddot{\Phi} = \Phi_{\mathbf{q}} \ddot{\mathbf{q}} + (\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} + 2\Phi_{\mathbf{q}t} \dot{\mathbf{q}} + \Phi_{tt} = \Phi_{\mathbf{q}} \ddot{\mathbf{q}} - \Gamma = \mathbf{0} \Rightarrow \Phi_{\mathbf{q}} \ddot{\mathbf{q}} = \Gamma \quad (2.4)$$

Eventually, Eqs. (2.1) and (2.4) can be written as index-1 DAEs (de Jalón and Bayo, 1994; Haug, 1989)

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^e \\ \Gamma \end{bmatrix} \quad (2.5)$$

where the coefficient matrix is called the augmented matrix (Negrut *et al.*, 1997; de Jalón and Gutiérrez-López, 2013). It should be pointed out that Eqs. (2.2), (2.3) and (2.4) are analytically equivalent. However, the direct numerical solution of Eq. (2.5) does not often provide the fulfillment of position (2.2) and velocity (2.3) constraints and may cause the solution drift. Therefore,

the stabilization methods are often employed. One of the simplest is the Baumgarte stabilization method (Baumgarte, 1972). In this method, the $\mathbf{\Gamma}$ vector from acceleration constraints is replaced by the following expression: $\bar{\mathbf{\Gamma}} = \mathbf{\Gamma} - 2\hat{\alpha}\dot{\mathbf{\Phi}} - \hat{\beta}^2\mathbf{\Phi}$, where $\hat{\alpha}$, $\hat{\beta}$ are the Baumgarte stabilization parameters. The value of these parameters is often assumed as $\hat{\alpha} = \hat{\beta} \in \langle 1, 20 \rangle$ (de Jalón and Bayo, 1994).

It is also possible to stabilize the system using the well-known Newton-Raphson method (Frączek and Wojtyra, 2008; Haug, 1989). The \mathbf{q} vector is corrected when constraints norm (2.2) is greater than the assumed tolerance.

3. Orthogonal complement methods

The equations of motion of the multibody system presented previously constitute a set of DAEs. In the present Section, those equations are transformed into ODEs by means of the orthogonal complement. To perform the transformation, the projection matrix \mathbf{P} that is orthogonal to the constraint Jacobian matrix must be computed. The application of the \mathbf{P} matrix leads to direct removal of the Lagrange multipliers from the considered system (de Jalón and Bayo, 1994).

3.1. Orthogonal complement for rheonomic constraints

The following statements are derived for systems without redundant constraints and then generalized to the case of redundant multibody systems. The procedure for finding orthogonal complement matrices is presented in (de Jalón and Bayo, 1994; Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Pełal, 2012; de Jalón and Gutiérrez-López, 2013).

Assume that the independent velocity vector $\dot{\mathbf{v}}$ can be obtained by the projection of the generalized velocity vector into the rows of the constant matrix \mathbf{B} (de Jalón and Bayo, 1994; Pennestrì and Valentini, 2007)

$$\dot{\mathbf{v}}_{s \times 1} = \mathbf{B}_{s \times n} \dot{\mathbf{q}}_{n \times 1} \quad (3.1)$$

where s is the number of independent coordinates.

Combining Eqs. (3.1) and (2.3) yields (de Jalón and Bayo, 1994; Pennestrì and Valentini, 2007)

$$\begin{bmatrix} \mathbf{\Phi}_q \\ \mathbf{B} \end{bmatrix} \dot{\mathbf{q}} = \mathbf{X} \dot{\mathbf{q}} = \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{v}} \end{bmatrix} \quad (3.2)$$

If \mathbf{X} is nonsingular, the inversion of this matrix exists and can be presented in the form: $\mathbf{X}^{-1} = [\mathbf{S} \ \mathbf{P}]$, where \mathbf{P} is the projection matrix and \mathbf{S} is the matrix which contains the remaining columns. Hence (de Jalón and Bayo, 1994; Pennestrì and Valentini, 2007)

$$\dot{\mathbf{q}} = \mathbf{X}^{-1} \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{v}} \end{bmatrix} = [\mathbf{S}_{n \times m} \ \mathbf{P}_{n \times s}] \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{v}} \end{bmatrix} = -\mathbf{S}\mathbf{\Phi}_t + \mathbf{P}\dot{\mathbf{v}} \quad (3.3)$$

Orthogonality of the \mathbf{X} matrix gives two orthogonality conditions in the form (de Jalón and Gutiérrez-López, 2013; Pennestrì and Valentini, 2007)

$$(\mathbf{\Phi}_q)_{m \times n} \mathbf{P}_{n \times s} = \mathbf{0}_{m \times s} \quad \mathbf{B}_{s \times n} \mathbf{P}_{n \times s} = \mathbf{I}_{s \times s} \quad (3.4)$$

The projection matrix \mathbf{P} can be used to eliminate the Lagrange multipliers.

Differentiating Eq. (3.2) and using Eq. (2.4) yields (de Jalón and Bayo, 1994; Pennestrì and Valentini, 2007)

$$\mathbf{X} \ddot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Gamma} \\ \ddot{\mathbf{v}} \end{bmatrix} \Rightarrow \ddot{\mathbf{q}} = \mathbf{X}^{-1} \begin{bmatrix} \mathbf{\Gamma} \\ \ddot{\mathbf{v}} \end{bmatrix} = [\mathbf{S} \ \mathbf{P}] \begin{bmatrix} \mathbf{\Gamma} \\ \ddot{\mathbf{v}} \end{bmatrix} = \mathbf{S}\mathbf{\Gamma} + \mathbf{P}\ddot{\mathbf{v}} \quad (3.5)$$

Substituting Eq. (3.5) into Eq. (2.1) gives

$$\mathbf{M}(\mathbf{S}\boldsymbol{\Gamma} + \mathbf{P}\ddot{\mathbf{v}}) + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathbf{T}}\boldsymbol{\lambda} = \mathbf{Q}^e \quad (3.6)$$

In order to eliminate the Lagrange multipliers, left-multiplication of both sides of Eq. (3.6) by $\mathbf{P}^{\mathbf{T}}$ is required. Afterwards, the use of transposition of Eq. (3.4)₁ yields (de Jalón and Bayo, 1994; de Jalón and Gutiérrez-López, 2013)

$$\mathbf{P}^{\mathbf{T}}\mathbf{M}(\mathbf{S}\boldsymbol{\Gamma} + \mathbf{P}\ddot{\mathbf{v}}) + \mathbf{0}\boldsymbol{\lambda} = \mathbf{P}^{\mathbf{T}}\mathbf{Q}^e \Leftrightarrow \mathbf{P}^{\mathbf{T}}\mathbf{M}\mathbf{P}\ddot{\mathbf{v}} = \mathbf{P}^{\mathbf{T}}\mathbf{Q}^e - \mathbf{P}^{\mathbf{T}}\mathbf{M}\mathbf{S}\boldsymbol{\Gamma} \quad (3.7)$$

Assuming $\ddot{\mathbf{v}}_{s \times 1} = \mathbf{0}$, $\mathbf{S}\boldsymbol{\Gamma}$ can be obtained from Eq. (3.5) as (de Jalón and Bayo, 1994; Pennestrì and Valentini, 2007)

$$\mathbf{S}\boldsymbol{\Gamma} = \mathbf{X}^{-1} \begin{bmatrix} \boldsymbol{\Gamma} \\ \mathbf{0} \end{bmatrix} \quad (3.8)$$

The vector $\ddot{\mathbf{v}}$ is obtained from Eq. (3.7). Substituting this vector into Eq. (3.5) yields $\ddot{\mathbf{q}}$.

In the case of redundant systems, it is convenient to use the pseudoinverse matrix \mathbf{X}^+ instead of \mathbf{X}^{-1} . Therefore, the problem becomes an optimization task, whose result has the least square norm.

3.2. Derivation of the projection matrix

Methods that directly apply the projection matrix \mathbf{P} are introduced in the following.

3.2.1. Zero eigenvalue method

The first method uses the zero eigenvalue technique (Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Pełal, 2012; Walton and Steeves, 1969) which is based on the eigenvalue problem. The eigenvalue problem can be written in the form (FreeMat v4.1; Hartfiel, 2001)

$$\mathbf{A}\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda} \quad (3.9)$$

where the matrix \mathbf{A} is a symmetrical matrix, $\boldsymbol{\Lambda}_{n \times n} = \text{diag}(\Lambda_1, \Lambda_2, \dots, \Lambda_n)$ contains eigenvalues and $\boldsymbol{\Psi}_{n \times n} = [\boldsymbol{\Psi}_1 \ \boldsymbol{\Psi}_2 \ \dots \ \boldsymbol{\Psi}_n]$ is the orthogonal modal matrix which contains the eigenvectors.

In order to determine the projection matrix \mathbf{P} , the symmetric matrix \mathbf{A} is considered as (Walton and Steeves, 1969; Pennestrì and Valentini, 2007)

$$\mathbf{A}_{n \times n} = \boldsymbol{\Phi}_{\mathbf{q}}^{\mathbf{T}}\boldsymbol{\Phi}_{\mathbf{q}} \quad (3.10)$$

Using this matrix, the following expression is obtained (Walton and Steeves, 1969; Pennestrì and Valentini, 2007)

$$\boldsymbol{\Psi}_{n \times n}^{\mathbf{T}}(\boldsymbol{\Phi}_{\mathbf{q}}^{\mathbf{T}})_{n \times m}(\boldsymbol{\Phi}_{\mathbf{q}})_{m \times n}\boldsymbol{\Psi}_{n \times n} = \boldsymbol{\Lambda}_{n \times n} \quad (3.11)$$

where $\boldsymbol{\Lambda}$ is unique and contains non-negative eigenvalues. Assume that $\Lambda_1 \leq \Lambda_2 \leq \dots \leq \Lambda_n$, then the possible zero eigenvalues have the lowest indices. Moreover, the number of positive eigenvalues is equal to the rank r of the Jacobian matrix. Therefore, $s = n - r$ eigenvalues are equal to zero. These eigenvalues correspond to rigid body motion, so their number is equal to the number of degrees of freedom (DOFs) of the system. Consequently, the system has s independent and $r = n - s$ dependent coordinates.

Assuming (Walton and Steeves, 1969)

$$(\boldsymbol{\Phi}_{\mathbf{q}})_{m \times n}\boldsymbol{\Psi}_{n \times n} = \mathbf{D}_{m \times n} \quad (3.12)$$

Equation (3.11) can be written as (Walton and Steeves, 1969)

$$\mathbf{D}^T \mathbf{D} = \mathbf{\Lambda} \quad (3.13)$$

The order of the eigenvalues is opposite to the sequence form (Walton and Steeves, 1969), thus

$$\mathbf{D}_{m \times n} = \begin{bmatrix} \mathbf{0}_{m \times s} & \bar{\mathbf{D}}_{m \times (n-s)} \end{bmatrix} \quad (3.14)$$

hence (Pennestrì and Valentini, 2007)

$$\Phi_{\mathbf{q}_{m \times n}} \Psi_{n \times s}^1 = \mathbf{0}_{m \times s} \quad (3.15)$$

where $\Psi_{n \times s}^1$ is the matrix created from the first s columns of the modal matrix, i.e. columns which correspond to the zero eigenvalues.

Note that equation (3.15) satisfies first orthogonality condition (3.4)₁, hence: $\mathbf{P} = \Psi_{n \times s}^1$. Moreover, the modal matrix is orthogonal so condition (3.4)₂ is fulfilled for: $\mathbf{B} = \mathbf{P}^T$.

3.2.2. Schur decomposition method

The description of the Schur decomposition method can be found in (Golub and Loan, 1996; Hartfiel, 2001; Kincaid and Cheney, 2002; Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Peğal, 2012). This decomposition is based on the fact that every square matrix \mathbf{A} can be presented in the form (Golub and Loan, 1996)

$$\mathbf{U}^H \mathbf{A} \mathbf{U} = \mathbf{T} = \mathbf{\Lambda} + \mathbf{N} \quad (3.16)$$

where \mathbf{U} is a unitary matrix, \mathbf{U}^H denotes conjugate transposition of \mathbf{U} and \mathbf{T} is a block triangular matrix, and \mathbf{N} is strictly the upper triangular matrix.

Note that the orthogonal matrix is a particular case of the unitary matrix for real numbers, thus (Golub and Loan, 1996)

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{\Lambda} + \mathbf{N} \quad (3.17)$$

Assuming the \mathbf{A} according to Eq. (3.10) and using the fact that the \mathbf{A} is normal, it is possible to write (Golub and Loan, 1996)

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{\Lambda} \quad (3.18)$$

Note that this is analogous to (3.11). Hence, it is possible to follow as for the zero eigenvalue method described in Section 3.2.1. Eventually, the following condition is get (Pennestrì and Valentini, 2007)

$$(\Phi_{\mathbf{q}})_{m \times n} \mathbf{U}_{n \times s}^1 = \mathbf{0}_{m \times s} \quad (3.19)$$

where \mathbf{U}^1 is the submatrix of \mathbf{U} (selected analogously to Ψ^1). Therefore, first orthogonality condition (3.4)₁ is fulfilled when: $\mathbf{P} = \mathbf{U}_{n \times s}^1$, while second orthogonality condition (3.4)₂ is satisfied for: $\mathbf{B} = \mathbf{P}^T$. Since the Schur decomposition method is equivalent to the zero eigenvalue method, these approaches are treated as one in the following.

3.2.3. PUTD method

The Pseudo Upper Triangular Decomposition method has two variants and their description can be found in (Amirouche, 2006; Ider and Amirouche, 1988; Mariti *et al.*, 2011, 2010; Ostalczyk, 2008; Pennestrì and Valentini, 2007; Pełal, 2012). The first type uses the Householder transformation and the second uses the Gauss elimination.

The transformation matrix \mathbf{H} is defined first. Using this matrix, the $\Phi_{\mathbf{q}}^T$ matrix can be transformed into the upper trapezoidal form \mathbf{Z} (in particular into the upper triangular form) as (Amirouche, 2006; Ider and Amirouche, 1988)

$$\mathbf{H}_{n \times n}^T (\Phi_{\mathbf{q}}^T)_{n \times m} = \mathbf{Z}_{n \times m} \quad (3.20)$$

where \mathbf{H} matrix can be obtained using the Householder transformation (the method denoted as PUTD-H) or the Gauss elimination (the method denoted as PUTD-G).

The matrix orthogonal to the \mathbf{Z} is found in the following step. This matrix can be computed using the Gram-Schmidt orthogonalization of the upper-triangularized Jacobian matrix, which gives an identity matrix \mathbf{D} (Amirouche, 2006). After this, the submatrix of \mathbf{D} is taken as (Pennestrì and Valentini, 2007)

$$\mathbf{D}_{2(n-r) \times n} = \begin{bmatrix} \mathbf{0}_{(n-r) \times r} & \mathbf{I}_{(n-r) \times (n-r)} \end{bmatrix} \quad (3.21)$$

thus (Pennestrì and Valentini, 2007)

$$\mathbf{D}_{2(n-r) \times n} \mathbf{H}_{n \times n}^T (\Phi_{\mathbf{q}}^T)_{n \times m} = \mathbf{0}_{(n-r) \times m} \quad (3.22)$$

which after transposing gives

$$(\mathbf{D}_2 \mathbf{H}^T \Phi_{\mathbf{q}}^T)^T = \Phi_{\mathbf{q}} (\mathbf{D}_2 \mathbf{H}^T)^T = \Phi_{\mathbf{q}} (\mathbf{H} \mathbf{D}_2^T) = \mathbf{0}_{m \times (n-r)} \quad (3.23)$$

Comparing Eqs. (3.23) and (3.4)₁ yields: $\mathbf{P} = \mathbf{H} \mathbf{D}_2^T$, while (3.4)₂ is fulfilled when: $\mathbf{B} = \mathbf{P}^T$, what is consistent with Pennestrì and Valentini (2007).

Considering the redundant systems, Eq. (3.21) is fulfilled when the redundant constraints are at the end of vector (2.2). It is not the general case. This problem can be solved by the Gauss-Jordan elimination with partial pivoting of the transposed Jacobian matrix. Using this elimination, the position of the independent constraints in the Jacobian matrix is obtained. Thus, it is possible to move dependent constraints at the end of the constraints set. Moreover, the problem of redundant constraints can be also solved by the manual setting of the constraints in Eq. (2.2) during the preprocessing stage. However, this simple approach can be used only for small systems.

In the implementation, the method based on the Gauss-Jordan elimination is used to detect redundant constraints. The choice of the independent constraints is done only once at the beginning of simulation. It should be noted that in some cases the coordinate partition must be done more frequently due to the loss of independence of the chosen coordinates.

3.2.4. SVD method

The method using singular value decomposition is described in (Mani *et al.*, 1985; Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Pełal, 2012). It uses the decomposition which can be written in the following form (Pennestrì and Valentini, 2007)

$$(\Phi_{\mathbf{q}}^T)_{n \times m} = \mathbf{W}_{n \times n} \mathbf{L}_{n \times m} \mathbf{V}_{m \times m}^T \quad (3.24)$$

where the matrices \mathbf{W} and \mathbf{V} are orthogonal, and \mathbf{L} includes singular values of the transposed Jacobian matrix on its diagonal. Assume that m is the total number of the eigenvalues of the $\Phi_{\mathbf{q}}^T$

and the rank of the Jacobian matrix r is equal to the number of nonzero eigenvalues. Thus, the decomposition can be written as (Pennestrì and Valentini, 2007)

$$(\Phi_{\mathbf{q}}^T)_{n \times m} = \left[(\mathbf{W}_d)_{n \times r} \quad (\mathbf{W}_i)_{n \times (n-r)} \right] \begin{bmatrix} \Lambda_{r \times m} \\ \mathbf{0}_{(n-r) \times m} \end{bmatrix} \mathbf{V}_{m \times m}^T = \mathbf{W}_d \Lambda \mathbf{V}^T \quad (3.25)$$

Left-multiplication of Eq. (3.25) by \mathbf{W}_i^T yields (Pennestrì and Valentini, 2007)

$$\mathbf{W}_i^T (\Phi_{\mathbf{q}}^T)_{n \times m} = \mathbf{W}_i^T \mathbf{W}_d \Lambda \mathbf{V}^T \quad (3.26)$$

Using orthogonality of the matrix \mathbf{W} gives (Pennestrì and Valentini, 2007)

$$\mathbf{W}_i^T \mathbf{W}_d = \mathbf{0} \quad \mathbf{W}_i^T \mathbf{W}_i = \mathbf{I} \quad (3.27)$$

hence (Pennestrì and Valentini, 2007)

$$\mathbf{W}_i^T (\Phi_{\mathbf{q}}^T)_{n \times m} = \mathbf{0}_{(n-r) \times m} \Rightarrow \Phi_{\mathbf{q}} \mathbf{W}_i = \mathbf{0}_{m \times (n-r)} \quad (3.28)$$

Eventually, first orthogonality condition (3.4)₁ is met for (Pennestrì and Valentini, 2007): $\mathbf{P} = \mathbf{W}_i$ and second condition (3.4)₂ is fulfilled when (Pennestrì and Valentini, 2007): $\mathbf{B} = \mathbf{P}^T$.

3.2.5. QR decomposition method

The QR method is described in, e.g. (Kim and Vanderploeg, 1986; Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Peçkal, 2012). It is based on the QR decomposition described in details by Golub and Loan (1996). This decomposition takes the form (Kim and Vanderploeg, 1986)

$$\Phi_{\mathbf{q}}^T = \mathbf{Q} \mathbf{R} \quad (3.29)$$

where the matrix \mathbf{Q} is orthogonal and \mathbf{R} is upper trapezoidal or upper triangular in the particular case of the square decomposed matrix. The matrix \mathbf{R} can be written as (Pennestrì and Valentini, 2007)

$$\mathbf{R}_{n \times m} = \begin{bmatrix} (\mathbf{R}_1)_{r \times m} \\ \mathbf{0}_{(n-r) \times m} \end{bmatrix} \quad (3.30)$$

However, when the redundant systems are considered, this form of the \mathbf{R} matrix occurs only when the redundant constraints are placed at the end of constraints vector (2.2). There are several methods to achieve this, e.g. appropriate definition of the constraints vector, use of the Gauss-Jordan elimination with partial pivoting or application of the alternative version of the QR decomposition which can be written in the form (FreeMat v4.1; MATLAB®)

$$\Phi_{\mathbf{q}}^T \mathbf{E} = \mathbf{Q} \mathbf{R} \quad (3.31)$$

where \mathbf{E} is a permutation matrix, which allows the matrix \mathbf{R} to take the form from Eq. (3.30). Eventually, the QR decomposition of the transposed Jacobian matrix yields

$$(\Phi_{\mathbf{q}}^T)_{n \times m} \mathbf{E}_{m \times m} = \mathbf{Q}_{n \times n} \mathbf{R}_{n \times m} = \begin{bmatrix} \mathbf{Q}_1_{n \times r} & \mathbf{Q}_2_{n \times (n-r)} \end{bmatrix} \begin{bmatrix} \mathbf{R}_1_{r \times m} \\ \mathbf{0}_{(n-r) \times m} \end{bmatrix} = \mathbf{Q}_1 \mathbf{R}_1 \quad (3.32)$$

Left-multiplying Eq. (3.32) by the \mathbf{Q}_2^T gives

$$\mathbf{Q}_2^T \Phi_{\mathbf{q}}^T \mathbf{E} = \mathbf{Q}_2^T \mathbf{Q}_1 \mathbf{R}_1 \quad (3.33)$$

Using the orthogonality property of the \mathbf{Q} matrix yields (Pennestrì and Valentini, 2007)

$$\mathbf{Q}_2^T \mathbf{Q}_1 = \mathbf{0} \quad \mathbf{Q}_2^T \mathbf{Q}_2 = \mathbf{I} \quad (3.34)$$

Transposition of Eq. (3.33) using Eq. (3.34) gives

$$(\mathbf{Q}_2^T \Phi_q^T \mathbf{E})^T = \mathbf{E}^T \Phi_q \mathbf{Q}_2 = \mathbf{0} \quad (3.35)$$

Left-multiplying Eq. (3.35) by the permutation matrix yields

$$\Phi_q \mathbf{Q}_2 = \mathbf{0} \quad (3.36)$$

Thus, first orthogonality condition (3.4)₁ is fulfilled when: $\mathbf{P} = \mathbf{Q}_2$ and second orthogonality condition (3.4)₂ is met for: $\mathbf{B} = \mathbf{P}^T$.

3.3. Related methods

The following Section describes methods that do not use directly the projection matrix \mathbf{P} . Note that these methods are equivalent to the orthogonal complement methods. It was described by, e.g. de Jalón and Bayo (1994) in the case of the coordinate partitioning method and by Wang and Huston (1989) for the Wang-Huston formulation.

3.3.1. Coordinate partitioning method

The coordinate partitioning method is described in (de Jalón and Bayo, 1994; Mariti *et al.*, 2011, 2010; Pennestrì and Valentini, 2007; Pełal, 2012; Wehage and Haug, 1982).

The coordinates are partitioned into independent \mathbf{v} and dependent \mathbf{u} sets using, e.g. the Gauss-Jordan elimination with partial pivoting of the Jacobian matrix. Thus, according to the coordinate partitioning, the differential-algebraic equations of motion from Eq. (2.5) take the following form (these are dependences from Pennestrì and Valentini (2007), written in the matrix form)

$$\begin{bmatrix} \mathbf{M}^{uu} & \mathbf{M}^{uv} & \Phi_u^T \\ \mathbf{M}^{vu} & \mathbf{M}^{vv} & \Phi_v^T \\ \Phi_u & \Phi_v & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{v}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{Q}^{eu} \\ \mathbf{Q}^{ev} \\ \Gamma \end{bmatrix} \quad (3.37)$$

giving

$$\begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{v}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \Phi_u^{-1}(\Gamma - \Phi_v \check{\mathbf{M}}^{-1} \check{\mathbf{Q}}^e) \\ \check{\mathbf{M}}^{-1} \check{\mathbf{Q}}^e \\ (\Phi_u^{-1})^T(\mathbf{Q}^{eu} - \mathbf{M}^{uv} \check{\mathbf{M}}^{-1} \check{\mathbf{Q}}^e - \mathbf{M}^{uu} \Phi_u^{-1}(\Gamma - \Phi_v \check{\mathbf{M}}^{-1} \check{\mathbf{Q}}^e)) \end{bmatrix} \quad (3.38)$$

where (Pennestrì and Valentini, 2007)

$$\begin{aligned} \check{\mathbf{M}} &= \mathbf{M}^{vv} - \mathbf{M}^{vu} \Phi_u^{-1} \Phi_v - \Phi_v^T (\Phi_u^{-1})^T (\mathbf{M}^{uv} - \mathbf{M}^{uu} \Phi_u^{-1} \Phi_v) \\ \check{\mathbf{Q}}^e &= \mathbf{Q}^{ev} - \mathbf{M}^{vu} \Phi_u^{-1} \Gamma - \Phi_v^T (\Phi_u^{-1})^T (\mathbf{Q}^{eu} - \mathbf{M}^{uu} \Phi_u^{-1} \Gamma) \end{aligned} \quad (3.39)$$

Note that the Φ_u^{-1} should be replaced by its pseudoinverse Φ_u^+ for redundant systems. Moreover, all the generalized coordinates are integrated during the simulations in order to obtain comparable results to the outcomes from the other methods.

3.3.2. Wang-Huston formulation

The method, described by Mariti *et al.* (2011, 2010), Pečkal (2012), Wang and Huston (1989) is based on pseudoinversion and is equivalent to the orthogonal complement method (Wang and Huston, 1989).

Using Eq. (2.1) yields (Mariti *et al.*, 2010; Wang and Huston, 1989)

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{Q}^e - \Phi_{\mathbf{q}}^T \lambda) \quad (3.40)$$

Substituting Eq. (3.40) into Eq. (2.4) gives (Wang and Huston, 1989)

$$\lambda = (\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^{-1} (\Phi_{\mathbf{q}} \mathbf{M}^{-1} \mathbf{Q}^e - \Gamma) \quad (3.41)$$

and substituting Eq. (3.41) into Eq. (3.40) leads to (Wang and Huston, 1989)

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T (\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^{-1} \Gamma - (\mathbf{M}^{-1} \Phi_{\mathbf{q}}^T (\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^{-1} \Phi_{\mathbf{q}} - \mathbf{I}) \mathbf{M}^{-1} \mathbf{Q}^e \quad (3.42)$$

where $\Phi_{\mathbf{q}}^T (\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^{-1}$ is the weighted pseudoinverse of the $\Phi_{\mathbf{q}}$ matrix (Wang and Huston, 1989).

For the redundant systems, the pseudoinverse matrix $(\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^+$ should be taken instead of the inverse matrix $(\Phi_{\mathbf{q}} \mathbf{M}^{-1} \Phi_{\mathbf{q}}^T)^{-1}$.

It is important to note that the Wang-Huston formulation is not suitable for multibody systems with singular mass matrices. Therefore, in the numerical examples presented below, the Euler angles are used in order to avoid this issue.

4. Numerical examples

4.1. McPherson suspensions with and without redundant constraints

The presented methods have been implemented and tested on two elementary examples of spatial mechanisms. The simplified McPherson strut without redundant constraints shown in Fig. 1a is the first example and the second one is the overconstrained McPherson strut presented in Fig. 1b. Elimination of the redundant constraints from the second example results in the simplified McPherson strut. The idea for consideration of these mechanisms is taken from (Haug, 1989). Both mechanisms consist of 5 rigid bodies (denoted below as $i = 1, 2, 3, 4, 5$) and have 4 degrees of freedom. Two of the DOFs are local mobilities (for bodies 3 and 4). Dimensions of the systems are presented in Table 1 and employ symbols presented in Figs. 1a and 1b. Moreover, origins of the local coordinate frames are located in the centers of mass of the bodies, so that certain expressions get simplified, e.g. the gravitational torques reduce to zero. It is assumed that the centers of mass are placed in the middle of the body i lengths: $|AB|$, $|CD|$, $|JK|$, $|IL|$ and $|LN|$, respectively.

The kinematics of the mechanisms is described using the absolute coordinates. The Euler angles are used for description of the orientation in order to obtain nonsingular mass matrices. Masses of all bodies are: $m_i = 1$ kg and moments of inertia are: $J_{xi} = J_{yi} = J_{zi} = 0.1$ kgm².

Table 1. Dimensions of the McPherson struts

Body	1			2						3	4	5	
Symbol	$ AB $	$ AA_a $	$ AA_b $	$ CD $	$ CE $	$ CF $	$ BC $	$ FG $	$ EH $	$ GI $	$ JK $	$ IL $	$ LN $
Dim. [m]	0.3	0.025	0.025	0.45	0.1	0.25	0.05	0.1	0.15	0.05	0.3	0.5	0.2

Both considered models are loaded in the same manner. The force of the constant value $F_z = 40$ N is applied to the center of mass of the first body in the z direction and the time

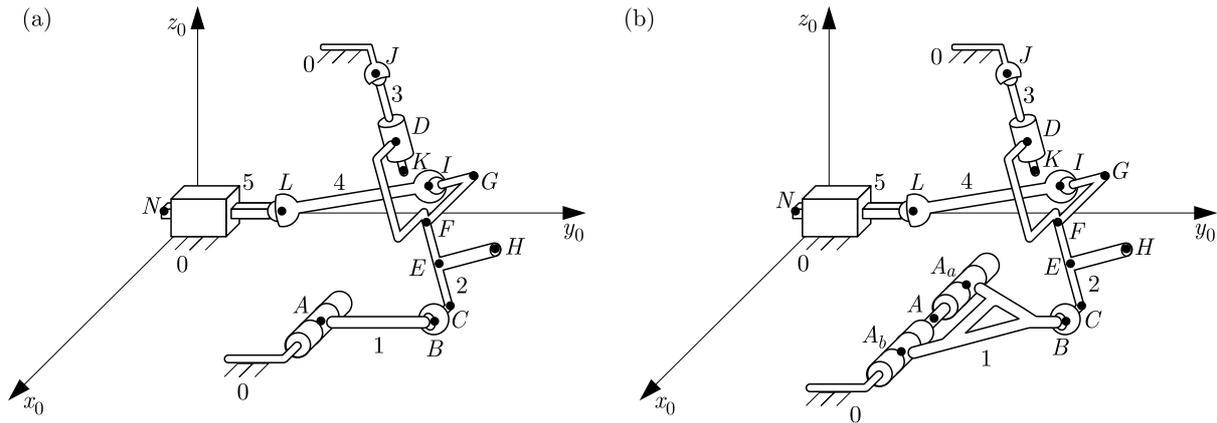


Fig. 1. (a) Simplified McPherson strut, (b) McPherson strut with redundant constraints

varying force acting in the y direction $F_y = 0.1 \sin(t) \text{ N}$ is applied to body 5. Moreover, it is assumed that the gravity is acting in the negative z direction with gravity acceleration equal to $g = -9.80665 \text{ m/s}^2$.

The analyzed methods have been implemented using MATLAB[®] R2012b. The obtained results (positions, velocities and accelerations) were compared with the outcomes of the simulations performed in Adams[™] 2013 in order to verify their correctness. To integrate the equations of motion, the ode45 method based on the Runge-Kutta scheme was used in MATLAB. Simulations were performed on the personal 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. Computational times were measured using *tic* and *toc* functions. Moreover, MATLAB program was run in the single thread mode in order to avoid problems with multi thread time measurements.

Each simulation was carried out for 10s motion. Three stabilization cases were performed for each method: without constraint stabilization, with the Baumgarte stabilization (where $\hat{\alpha} = \hat{\beta} = 10$) and with the stabilization of the position constraints using the Newton-Raphson method. Moreover, four ode45 error tolerances were considered: $AbsTol = RelTol \in \{1e-10; 1e-8; 1e-6; 1e-3\}$, where *AbsTol* and *RelTol* are absolute and relative error tolerances respectively.

4.2. Numerical results – comparison

Results of the computational time versus the error tolerances are presented in Figs. 2-5. Figures 2a and 3a contain the computation time for simulations without constraint stabilization, Figs. 2b and 3b present the results for computations with the Baumgarte stabilization, and Figs. 4 and 5 depict outcomes from simulations with stabilization of the position constraints. Note that Fig. 5b shows the results presented in Fig. 5a but for clarity the Wang-Huston formulation results are excluded.

In the most examples, PUTD methods turn out to be the slowest. Note that the PUTD-H and PUTD-G give almost the same computational time in the case of the simplified McPherson strut, while for the overconstrained McPherson strut, the PUTD-G seems to be faster than the PUTD-H. It may be due to inefficient implementation of the transformation of the \mathbf{H} matrix (Eq. (3.20)). For the remaining methods, the results of most simulation cases are very close to each other. However, in the case of the redundant McPherson strut with stabilization of the position constraints, the Wang-Huston formulation is many times slower than other methods, because of the loss of accuracy of the velocity constraints. Hence, the QR decomposition can be recognized as the most reliable and efficient method though, its results are a bit slower than the outcomes from the Wang-Huston formulation in some cases.

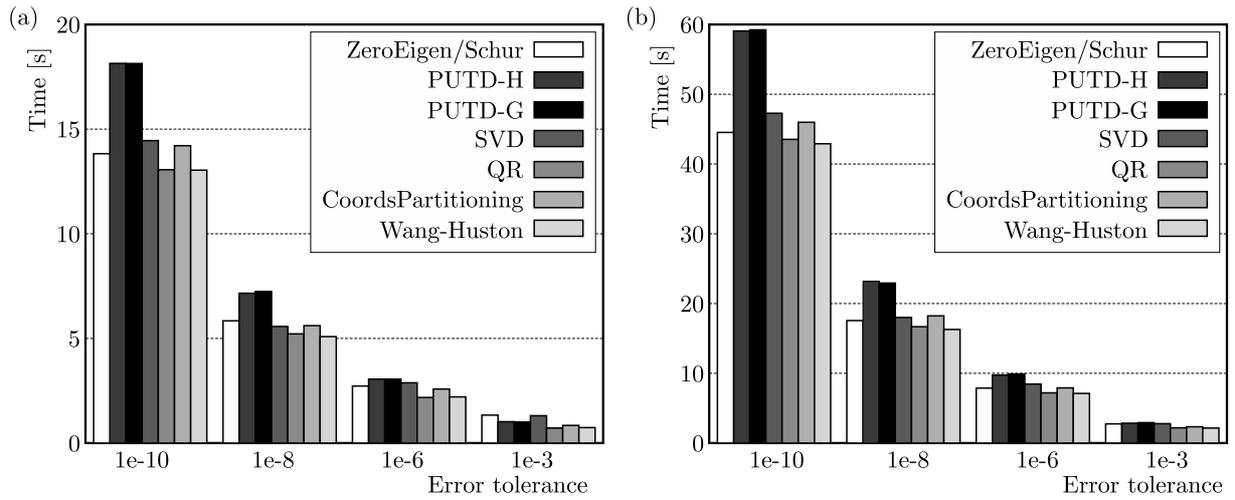


Fig. 2. Simplified McPherson strut: (a) without constraint stabilization, (b) with the Baumgarte stabilization

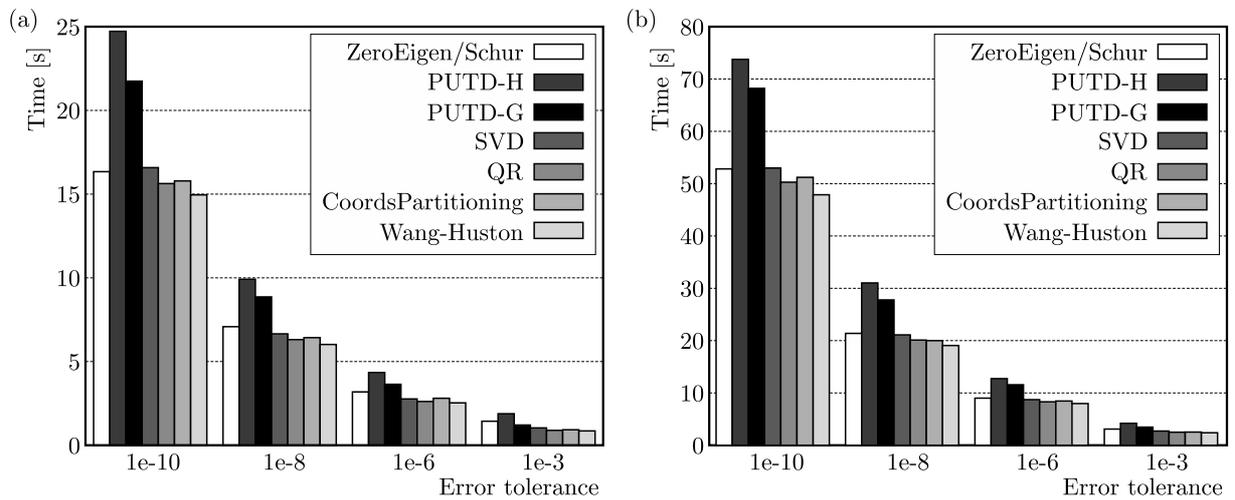


Fig. 3. Overconstrained McPherson strut: (a) without constraint stabilization, (b) with the Baumgarte stabilization

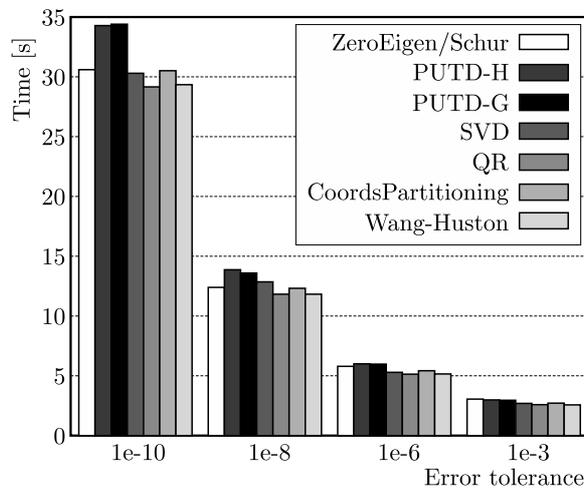


Fig. 4. Simplified McPherson strut with stabilization of the position constraints

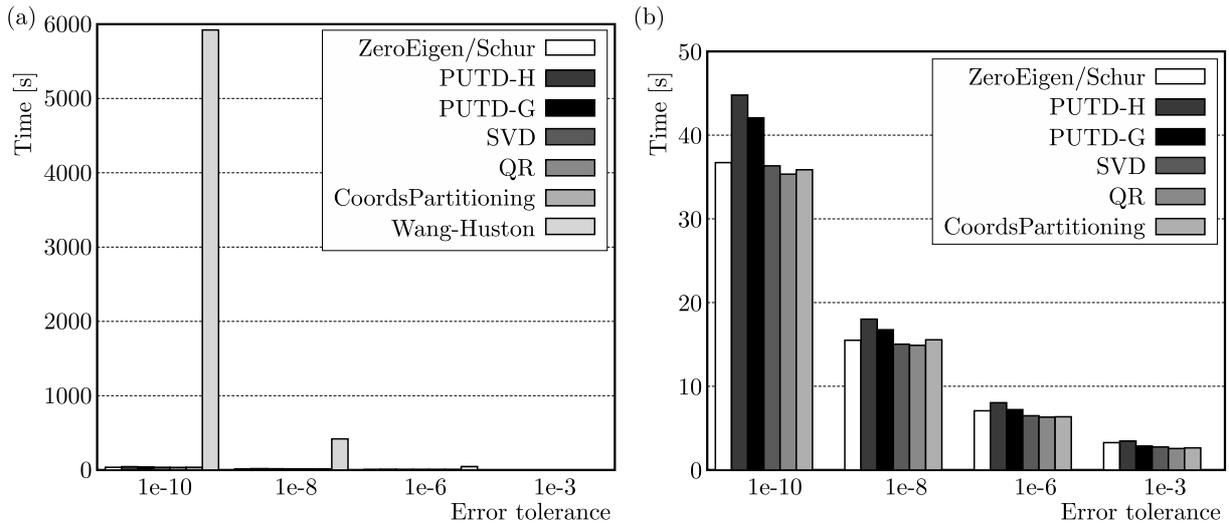


Fig. 5. Overconstrained McPherson strut with stabilization of the position constraints: (a) with and (b) without results of the Wang-Huston formulation

The simulations with stabilization of the constraints take longer than the simulations without stabilization, which should be expected due to greater computational cost.

There are other publications that compare the computation time for spatial systems, e.g. (Mariti *et al.*, 2011, 2010), where results for the Wang-Huston formulation are not presented (when the spatial systems are examined). This is because the Euler parameters were used there for description of the orientation, which resulted in the singular mass matrix and, consequently, the Wang-Huston formulation cannot be employed.

Moreover, the zero eigenvalue method and the Schur decomposition method were considered separately by Mariti *et al.* (2011, 2010), Pennestrì and Valentini (2007) but the matrix \mathbf{A} (Eq. (3.10)) is symmetric, so the Schur decomposition is equivalent to the eigenvalue problem (Golub and Loan, 1996). Therefore, these methods can be considered together as it is done in the current paper. Furthermore, only one type of the PUTD method was examined by Mariti *et al.* (2011, 2010), and in our paper 2 variants of that method are analyzed.

Comparing to Mariti *et al.* (2010), the shorter qualitative results are obtained for the coordinate partitioning method. This is due to the fact that in Mariti *et al.* (2010) coordinate partitioning was done at each time step. In our paper, only one coordinate partitioning is sufficient. In the general case, it should be monitored whether the actual partition of coordinates is still valid.

The computation time grows with the increasing accuracy of computations what is in general consistent with the intuition. This is not the case for all results presented by Mariti *et al.* (2011).

As mentioned earlier, the results of the zero eigenvalue, Schur decomposition, SVD and QR decomposition methods are close to each other. The similar conclusion was obtained by Mariti *et al.* (2010). Note also that in Mariti *et al.* (2010), the PUTD method gave results that were close to the mentioned methods. This is not the case in the present work as computations for both types of the PUTD method usually take longer than for the zero eigenvalue, Schur decomposition, SVD or QR decomposition methods. This might be caused by inefficient implementation of the \mathbf{H} matrix transformation (Eq. (3.20)).

5. Conclusions

Computational effectiveness of the simulation algorithms strongly depends on the selected method. In the current paper methods that are based on the orthogonal complement are compared:

zero eigenvalue formulation, Pseudo Upper Triangular Decomposition (PUTD), Schur decomposition, Singular Value Decomposition (SVD) and QR decomposition. Moreover, two schemes equivalent to the orthogonal complement methods are also considered: coordinate partitioning and Wang-Huston formulation. The effectiveness of these methods for two elementary mechanisms – McPherson struts with and without redundant constraints are considered. Moreover, it is worth noting that the obtained results are comparable with the outcomes from the previous publications (Mariti *et al.*, 2010; 2011). The comparison with the mentioned publications is described in details in Section 4.

The most robust and one of the most efficient formulation is the method based on the QR decomposition, although Wang-Huston formulation is faster in some cases. This is because the Wang-Huston formulation proved to be the slowest method in simulations of the redundant mechanism with stabilization of the position constraints. It is due to the loss of accuracy of the velocity constraints, which are not stabilized in that case. The other algorithms can be slower because of several reasons, e.g. they may require the solution of the eigenvalue problem, which is numerically expensive. Moreover, the computation time is shorter for simulations without stabilization than for the cases with stabilization. Despite this, there is a decrease in the constraints accuracy in the unstabilized simulations, which may cause difficulties in longer multibody motion analyses.

Acknowledgements

This work has been supported by the National Science Centre of Poland under grant No. DEC-2012/07/B/ST8/03993.

References

1. AMIROUCHE F.M.L., 2006, *Fundamentals of Multibody Dynamics, Theory and Applications*, Birkhäuser Boston
2. AWREJCEWICZ J., 2014, *Ordinary Differential Equations and Mechanical Systems*, Springer International Publishing
3. AWREJCEWICZ J., KUDRA G., 2005, Modeling, numerical analysis and application of triple physical pendulum with rigid limiters of motion, *Archive of Applied Mechanics*, **74**, 11-12, 746-753
4. AWREJCEWICZ J., KUDRA G., 2014, Mathematical modelling and simulation of the bifurcational wobblestone dynamics, *Discontinuity, Nonlinearity, and Complexity*, **3**, 2, 123-132
5. AWREJCEWICZ J., KUDRA G., LAMARQUE C.H., 2003, Dynamics investigation of three coupled rods with a horizontal barrier, *Meccanica*, **38**, 6, 687-698
6. AWREJCEWICZ J., KUDRA G., LAMARQUE C.H., 2004, Investigation of triple pendulum with impacts using fundamental solution matrices, *International Journal of Bifurcation and Chaos*, **14**, 12, 4191-4213
7. BAUMGARTE J.W., 1972, Stabilization of constraints and integrals of motion in dynamical systems, *Computer Methods in Applied Mechanics and Engineering*, **1**, 1, 1-16
8. DE JALÓN J.G., BAYO E., 1994, *Kinematic and Dynamic Simulation of Multibody Systems, The Real-Time Challenge*, Springer-Verlag, New-York
9. DE JALÓN J.G., GUTIÉRREZ-LÓPEZ M.D., 2013, Multibody dynamics with redundant constraints and singular mass matrix: existence, uniqueness, and determination of solutions for accelerations and constraint forces, *Multibody System Dynamics*, **30**, 3, 311-341
10. EICH-SOELLNER E., FÜHRER C., 1998, *Numerical Methods in Multibody Dynamics*, B.G. Teubner
11. FRĄCZEK J., WOJTYRA M., 2008, *Kinematyka układów wieloczłonowych. Metody obliczeniowe*, Wydawnictwa Naukowo-Techniczne, Warszawa

12. FreeMat v4.1, help
13. GOLUB G.H., LOAN C.F.V., 1996, *Matrix Computations*, The Johns Hopkins University Press, 3rd edition
14. HARTFIEL D.J., 2001, *Matrix Theory and Applications with MATLAB*, CRC Press
15. HAUG E.J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods*, Allyn and Bacon
16. IDER S.K., AMIROUCHE F.M.L., 1988, Coordinate reduction in the dynamics of constrained multibody systems – a new approach, *ASME Journal of Applied Mechanics*, **55**, 4, 899-904
17. KIM S.S., VANDERPLOEG M.J., 1986, QR decomposition for state space representation of constrained mechanical dynamic systems, *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **108**, 2, 183-188
18. KINCAID D., CHENEY W., 2002, *Numerical Analysis: Mathematics of Scientific Computing*, Brooks/Cole, 3-rd edition
19. KUNKEL P., MEHRMAN V., 2006, *Differential-Algebraic Equations*, European Mathematical Society
20. MALCZYK P., FRĄCZEK J., 2012, A divide and conquer algorithm for constrained multibody system dynamics based on augmented Lagrangian method with projections-based error correction, *Nonlinear Dynamics*, **70**, 1, 871-889
21. MANI N.K., HAUG E.J., ATKINSON K.E., 1985, Application of singular value decomposition for analysis of mechanical system dynamics, *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, **107**, 1, 82-87
22. MARITI L., BELFIORE N.P., PENNESTRÌ E., VALENTINI P.P., 2011, Comparison of solution strategies for multibody dynamics equations, *International Journal for Numerical Methods in Engineering*, **88**, 7, 637-656
23. MARITI L., PENNESTRÌ E., VALENTINI P.P., BELFIORE N.P., 2010, Review and comparison of solution strategies for multibody dynamics equations, *The 1st Joint International Conference on Multibody System Dynamics*, Lappeenranta, Finland
24. MATLAB®, help
25. NEGRUT D., SERBAN R., POTRA F.A., 1997, A topology-based approach to exploiting sparsity in multibody dynamics: joint formulation, *Mechanics of Structures and Machines*, **25**, 2, 221-241
26. OSTALCZYK P., 2008, *Wybrane zagadnienia rachunku wektorowego i macierzowego dla robotyków*, Wydawnictwo Politechniki Łódzkiej, Łódź
27. PENNESTRÌ , VALENTINI P.P., 2007, Coordinate reduction strategies in multibody dynamics: A review, *Proceedings of the Conference on Multibody System Dynamics*
28. PĘKAL M., 2012, *Porównanie metod całkowania RRA w analizie dynamiki układów wieloczłonowych* (in Polish), praca dyplomowa magisterska, Politechnika Warszawska, Warszawa
29. WALTON W.C., STEEVES E.C., 1969, A New Matrix Theorem and Its Application for Establishing Independent Coordinates for Complex Dynamical Systems with Constraints, NASA Technical Report: NASA TR R-326, NASA, Washington D. C.
30. WANG J.T., HUSTON R.L., 1989, A comparison of analysis methods of redundant multibody systems, *Mechanics Research Communications*, **16**, 3, 175-182
31. WEHAGE R.A., HAUG E.J., 1982, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, *ASME Journal of Mechanical Design*, **104**, 1, 247-255
32. WOJTYRA M., FRĄCZEK J., 2013, Comparison of selected methods of handling redundant constraints in multibody systems simulations, *Journal of Computational and Nonlinear Dynamics*, **8**, 2, 021007 (1-9)