P. Zhou, H. Ren, P. Masarati
A Relaxed Coupling Method for Algebraically Constrained Mechanical Systems
Multibody System Dynamics, Published online 11/05/2022
doi:10.1007/s11044-022-09825-0

A relaxed coupling method for algebraically constrained mechanical systems[1]

Ping Zhou*, Hui Ren+, Pierangelo Masarati†

* School of Astronautics, Harbin Institute of Technology,
Harbin, P. R. China, pingzhou@stu.hit.edu.cn
+ Institute of Aerospace Vehicle Dynamics and Control,
School of Astronautics, Harbin Institute of Technology,
Harbin, P. R. China, renhui@hit.edu.cn
† Department of Aerospace Science and Technology,
Politecnico di Milano, via La Masa 34,
Milan, Italy, pierangelo.masarati@polimi.it

Abstract

A coupling method is presented that aims at computing the dynamics of constrained mechanical systems connected by algebraic constraints. A relaxed coupling method is proposed, where each subsystem is reformulated as a set of ODEs and solved with an iteration process. The method is straightforward to implement, also for parallelization. The core idea is to eliminate the Lagrange multipliers of the DAEs that describe the constraint dynamics of each subsystem using a proper constraint stabilization technique. A linear combination of the constraint equations at position and velocity level is enforced, to prevent the occurrence of numerical drifting. The associated stabilization parameter is chosen in relation to the time step size. The effectiveness of the proposed approach is verified by solving a three-dimensional problem with rigid and flexible bodies. The results show that the method is effective in co-simulating algebraically constrained mechanical systems.

Keywords: constrained mechanical systems; flexible systems; algebraic constraints; ordinary differential equations; co-simulation

1 Introduction

Coupling techniques, also called co-simulation, have been successfully developed in the past decades; for an accurate description of the state of the art, one can refer to [1–3]. Co-simulation approaches have been widely applied to multidisciplinary problems, such as fluid/structure interaction [4–7], vehicle dynamics [8–10],

[1] Multibody System Dynamics, https://doi.org/10.1007/s11044-022-09825-0

coupled multibody and hydraulic systems [11, 12], and coupled particle model and multibody system dynamics [13, 14]. Co-simulation approaches are also successfully used in mono-disciplinary areas [15–21]. In the field of multibody systems, co-simulation is often used to parallelize computations, optimizing the use of computer resources and reducing the complexity of the problem [22].

Recently, many researchers focused on algorithms for coupling already distributed subsystems, either by algebraic constraint equations [17, 23–28] or by constitutive equations [20,29–32]. In addition, in terms of the computation sequence of the subsystem integration, the coupling schemes can either be parallel, namely the Jacobi scheme, or sequential, such as Gauss-Seidel's. The basic idea of coupling (or gluing) distributed subsystems is to exchange information at the interface of the subsystems. In this context, information is either one of the displacements, velocities, and accelerations of the interface or the action-reaction forces at the interface. Such information is subsequently updated iteratively to satisfy coupling conditions, which are either equilibrium or compatibility conditions. The advantage of this perspective is that only interface information is transmitted between the coupled subsystems, preserving the independence of each subsystem. Drawback are the potential loss of algorithmic stability and the complexity of the algorithms themselves. For example, [27] reports that "... the implementation of the method is more complicated [...] the time step size for the subsystem integration has to be chosen properly to correctly calculate the coupling Jacobian".

The methods above are successfully applied to the co-simulation of rigid systems in MBD [23–32]. In contrast, investigations on co-simulating an MBD system with flexible bodies are rarely found in the literature; some relevant research can be found in [24, 35]. In flexible systems, if flexible joints are in the coupling condition among DAE subsystems, which is usually necessary with the gluing co-simulation technique, the algorithm will likely fail to converge in practical applications. In [24], for a planar four-bar mechanism with one flexible bar, the equations of motion of both subsystems are first reduced to ODEs before applying the above gluing co-simulation technique. In [35], in the case of an elastic multibody chain, the effects of flexible joints are incorporated into differential equations with an explicit solution of joint constraint forces. Besides, although planar MBD systems were frequently studied in the field of co-simulation, research on spatial problems is less frequent; some can be found in the literature [36, 37]. Flexible mechanical systems, which exhibit more complexity than planar ones in terms of both rotation description and deformation of flexible bodies, received less attention; relevant research is discussed for example in [38].

Here, an alternative perspective is considered to handle a general spatial MBD system with rigid and flexible bodies. This alternative is to partition the whole system into several subsystems and exchange the information of generalized coordinates and velocities at the communication macro-time nodes in an iterative process during each time step. The strategy takes advantage of parallel computing that enables large scale multibody systems to be simulated with ready-made parallel algorithms and machines. In this light, research was done in the past decades [33–35,39,40]. In particular, the waveform relaxation (WR) method, which was originally applied to the parallelization of large-scale integrated circuits in [41], was extended by Leimkuhler to solve DAEs of constrained mechanical systems [39]. For a pair of multibody systems, the governing equations of subsystem i $(i = 1, 2)$ are

$$\begin{gathered}\mathbf{M}_i \ddot{\boldsymbol{q}}_i + \boldsymbol{c}_{i\boldsymbol{q}_i}^T \boldsymbol{\lambda}_i = \boldsymbol{F}_i \\ \boldsymbol{c}_i = \mathbf{0}\end{gathered}$$

where $\mathbf{M}$ is the mass matrix, $\boldsymbol{q}$ is the vector of the generalized coordinates, $\boldsymbol{F}$ is the vector of the generalized forces, $\boldsymbol{\lambda}$ is the vector of the Lagrange multipliers, and $\boldsymbol{c}$ denotes the vector of the constraint equations. If two subsystems are coupled through force terms $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$, applying the WR iteration leads to

$$\begin{gathered}\mathbf{M}_1^{k+1} \ddot{\boldsymbol{q}}_1^{k+1} + \boldsymbol{c}_{1\boldsymbol{q}_1}^T \boldsymbol{\lambda}_1 = \boldsymbol{F}_1(t, \boldsymbol{q}_1^{k+1}, \dot{\boldsymbol{q}}_1^{k+1}, \boldsymbol{q}_2^k, \dot{\boldsymbol{q}}_2^k) \\ \boldsymbol{c}_1(\boldsymbol{q}_1^{k+1}, t) = \mathbf{0}\end{gathered} \tag{1}$$

and

$$\begin{gathered}\mathbf{M}_2^{k+1} \ddot{\boldsymbol{q}}_2^{k+1} + \boldsymbol{c}_{2\boldsymbol{q}_2}^T \boldsymbol{\lambda}_2 = \boldsymbol{F}_2(t, \boldsymbol{q}_2^{k+1}, \dot{\boldsymbol{q}}_2^{k+1}, \boldsymbol{q}_1^k, \dot{\boldsymbol{q}}_1^k) \\ \boldsymbol{c}_2(\boldsymbol{q}_2^{k+1}, t) = \mathbf{0}\end{gathered} \tag{2}$$

where k is the iteration counter. To advance one time step forward, iterations are mandatory. During each iteration, the coupling information of the subsystem comes from the previous results of the other subsystem. This method was demonstrated feasible for subsystems coupled by constitutive equations. However, for subsystems coupled through constraint equations, a case of interest in many multibody problems, this strategy of splitting systems at constraint interfaces while applying the WR method will fail to converge. As recommended by Tseng and Hulbert in [23], one remedy could be to replace DAEs with ODEs before applying WR. Following this idea, if the subsystems' dynamics can be expressed as ODEs, applying a WR iteration method will yield

$$\mathbf{M}_1^{k+1} \ddot{\boldsymbol{q}}_1^{k+1} = \boldsymbol{Q}_1(t, \boldsymbol{q}_1^{k+1}, \dot{\boldsymbol{q}}_1^{k+1}, \boldsymbol{q}_2^k, \dot{\boldsymbol{q}}_2^k) \tag{3}$$

and

$$\mathbf{M}_2^{k+1} \ddot{\boldsymbol{q}}_1^{k+1} = \boldsymbol{Q}_2(t, \boldsymbol{q}_2^{k+1}, \dot{\boldsymbol{q}}_2^{k+1}, \boldsymbol{q}_1^k, \dot{\boldsymbol{q}}_1^k) \tag{4}$$

Along these lines, a potentially applicable method is proposed in the present work.

The proposed co-simulation method is based on a set of ODEs and employs ODE integration schemes to conduct a relaxed coupling process based on the Gauss-Seidel scheme. Constrained mechanical systems with rigid and flexible bodies are considered here, formulated as index-3 DAEs. The Lagrange multipliers in the DAEs are eliminated by adopting a constraint stabilization technique. The numerical result is free from drift, since the constraint equations at position and velocity level are enforced through stabilization. Also, the mass matrices of the constrained systems in both rigid and flexible cases can be assumed to be constant during the co-simulation, such that they can be calculated and stored beforehand. Another advantage is that when implicit integration schemes are used, a numerically evaluated Jacobian matrix can often replace the analytical one without impacting the accuracy of the solution, since the value of the stabilization parameter is related to the time step size. For complex three-dimensional problems, where the derivation of the analytical Jacobian matrix can be highly intricate, this benefit is of particular significance. Although the simplicity of implementation with good numerical behaviour is achieved, the proposed method also suffers from some drawbacks. The method requires the iterative update of the variables exchanged between subsystems, which may not be possible in co-simulation environments that do not allow rollback.

The paper is structured as follows: Section 2 presents a review of the governing equations for constrained rigid and flexible bodies based on the Lie group methodology. Stabilized index-2 DAEs are correspondingly obtained, using the algorithm proposed by Gear et al. [42]. Only holonomic constraints are considered. Section 3 illustrates the derivation of ODEs from index-2 DAEs using a constraint stabilization technique. In Section 4, a monolithic system is partitioned into ODE subsystems, and the co-simulation implementation details are given. Section 5 presents three-dimensional numerical examples. Finally, conclusions are drawn in Section 6.

2 Problem Formulation

This section describes the formulation of the constrained dynamics of rigid and flexible bodies. The well-known Floating Frame of Reference Formulation (FFRF) and the Absolute Nodal Coordinate Formulation (ANCF) [45] are used in the latter case. They are briefly recalled in the following, along with the formulation of the dynamics of a rigid body, to show how these modeling paradigms can be all cast within the frame of the proposed co-simulation approach.

2.1 Dynamics of a Rigid Body

The global position of an arbitrary point P on a rigid body is expressed as

$$\boldsymbol{r}_p = \boldsymbol{r}(t) + \mathbf{A}(t)\boldsymbol{u}_0$$

where $\boldsymbol{r}$ is the position vector of the center of mass and $\boldsymbol{u}_0$ is the position of point P in the reference coordinate system; $\mathbf{A}$ is the orientation matrix. Considering a matrix $\mathbf{R}$ constructed by

$$\mathbf{R} = \mathbf{A}^T(t_k)\mathbf{A}(t) \tag{5}$$

where time t is at the vicinity of discrete time node t_k, it is directly verified that

$$\{\mathbf{R}^T\mathbf{R} = \mathbf{I}, \quad \mathtt{det}(\mathbf{R}) = 1\} \quad \Rightarrow \quad \mathbf{R} \in SO(3)$$

Then the matrix $\mathbf{R}$ can be parameterized by

$$\mathbf{R}(t) = \mathtt{exp}(\tilde{\boldsymbol{\theta}}) \tag{6}$$

where the operator $\tilde{(\cdot)}$ denotes the skew symmetric matrix constructed from the corresponding three-dimensional vector; $\boldsymbol{\theta}$ is the rotation increment from time t_k that defines the Lie group algebra $\tilde{\boldsymbol{\theta}}$; and the rotation increment $\exp(\tilde{\boldsymbol{\theta}})$, with

$$\exp(\tilde{\boldsymbol{\theta}}) = \mathbf{I}_3 + \frac{\sin(|\boldsymbol{\theta}|)}{|\boldsymbol{\theta}|}\tilde{\boldsymbol{\theta}} + \frac{1-\cos(|\boldsymbol{\theta}|)}{|\boldsymbol{\theta}|^2}\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}$$

Replacing the matrix $\mathbf{R}$ in Eq. (5) with the right hand side of Eq. (6) yields

$$\mathbf{A}(t) = \mathbf{A}(t_k)\exp(\tilde{\boldsymbol{\theta}}) \tag{7}$$

One can refer to [60] for related details.

The velocity of an arbitrary point P can be expressed as

$$\dot{\boldsymbol{r}}_p = \dot{\boldsymbol{r}} + \dot{\mathbf{A}}\boldsymbol{u}_0 = \dot{\boldsymbol{r}} - \mathbf{A}\tilde{\boldsymbol{u}}_0\boldsymbol{\omega}$$

where $\boldsymbol{\omega}$ is the angular velocity vector, resulting from the time derivative of the orientation matrix,

$$\tilde{\boldsymbol{\omega}} = \mathbf{A}^T\dot{\mathbf{A}} \tag{8}$$

The angular velocity can be expressed as a function of the time derivative of the rotation $\boldsymbol{\theta}$ according to

$$\boldsymbol{\omega} = \mathbf{G}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{9}$$

with

$$\mathbf{G}(\boldsymbol{\theta}) = \mathbf{I}_3 + \frac{1-\cos(|\boldsymbol{\theta}|)}{|\boldsymbol{\theta}|^2}\tilde{\boldsymbol{\theta}} + \frac{|\boldsymbol{\theta}| - \sin(|\boldsymbol{\theta}|)}{|\boldsymbol{\theta}|^3}\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}$$

Matrix $\mathbf{G}$ becomes singular for $|\boldsymbol{\theta}| = 2n\pi$ ($n \in \mathrm{N}^+$) in each time step. The singularity of matrix $\mathbf{G}(\boldsymbol{\theta})$ is avoided here as long as the incremental rotation is limited, namely $|\boldsymbol{\theta}| < 2\pi$, which is usually the case (actually, $|\boldsymbol{\theta}| \ll 2\pi$) when acceptable accuracy is sought.

Therefore, the kinetic energy of a rigid body can be written as

$$\mathscr{K} = \frac{1}{2}\int_V \rho\dot{\boldsymbol{r}}_p \cdot \dot{\boldsymbol{r}}_p dv = \dot{\boldsymbol{r}}^T \underbrace{\frac{1}{2}\int_V \rho\mathbf{I}_3 dv}_{\mathbf{m}}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{\omega}^T \underbrace{\int_V \rho\tilde{\boldsymbol{u}}_0^T\tilde{\boldsymbol{u}}_0 dv}_{\mathbf{J}}\boldsymbol{\omega} = \frac{1}{2}\dot{\boldsymbol{r}}^T\mathbf{m}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{\omega}^T\mathbf{J}\boldsymbol{\omega}$$

where the component of matrix $\mathbf{m}$ is the mass of the body, and the matrix $\mathbf{J}$ is the centroid principal axes of inertia moment. In analogy with the equality $\boldsymbol{\omega} = \mathbf{G}\dot{\boldsymbol{\theta}}$ of Eq. (9), the virtual rotation can be expressed as $\delta\boldsymbol{\pi} = \mathbf{G}\delta\boldsymbol{\theta}$. The virtual work done by an arbitrary force $\boldsymbol{f}$ and an arbitrary torque $\boldsymbol{t}$ is

$$\delta\mathscr{W} = \delta\boldsymbol{r}^T\boldsymbol{f} + \delta\boldsymbol{\pi}^T\boldsymbol{t} = \delta\boldsymbol{r}^T\boldsymbol{f} + \delta\boldsymbol{\theta}^T\mathbf{G}^T\boldsymbol{t}$$

Hence, the generalized forces with respect to the virtual displacements $\delta\boldsymbol{r}$ and $\delta\boldsymbol{\theta}$ are $\boldsymbol{f}$ and $\mathbf{G}^T\boldsymbol{t}$, respectively. The Lagrange equations of the first kind yield a set of DAEs for the constrained dynamical system

$$\mathbf{m}\ddot{\boldsymbol{r}} + \left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{r}}\right)^T\boldsymbol{\lambda} = \boldsymbol{f}, \qquad \mathbf{G}^T\{\mathbf{J}\dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega}\} + \left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{\theta}}\right)^T\boldsymbol{\lambda} = \mathbf{G}^T\boldsymbol{t} \tag{10}$$

where $\boldsymbol{c}$ denotes the constraint equation and is assumed to be holonomic for simplicity in this work; $\boldsymbol{\lambda}$ represents the related vector of Lagrange multipliers. Due to the equality $\dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\theta}} = \dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}\dfrac{\partial\boldsymbol{\pi}}{\partial\boldsymbol{\theta}} = \dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}\mathbf{G}$, substituting it in the second of Eqs. (10) and collecting matrix $\mathbf{G}^T$ yields

$$\mathbf{G}^T\left\{\mathbf{J}\dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} + \left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}\right)^T\boldsymbol{\lambda}\right\} = \mathbf{G}^T\boldsymbol{t}$$

Since matrix $\mathbf{G}$ is a function of a relative rotation vector $\boldsymbol{\theta}$, as in Eq. (9), it will not become singular as long as $|\boldsymbol{\theta}| < 2\pi$. Thus, premultiplication by matrix $\mathbf{G}^T$ can be eliminated from both sides of the equation without affecting the solution. It is worth noticing that this operation transforms the equilibrium of generalized forces, energetically conjugated with the virtual perturbation of the rotation parameters $\delta\boldsymbol{\theta}$, into true moment equilibrium equations, energetically conjugated with the virtual rotations $\delta\boldsymbol{\pi}$. In this way, the governing equations of a rigid body can be expressed by

$$\begin{gathered} \mathbf{m}\ddot{\boldsymbol{r}} + \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{r}}\right)^T \boldsymbol{\lambda} = \boldsymbol{f} \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} + \left(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\pi}}\right)^T \boldsymbol{\lambda} = \boldsymbol{t} \\ \boldsymbol{c}(t, \boldsymbol{r}, \mathbf{A}) = \mathbf{0} \end{gathered} \tag{11}$$

Denoting $\boldsymbol{q} = \left(\boldsymbol{r}^T, \boldsymbol{\theta}^T\right)^T$ and $\boldsymbol{v} = \left(\dot{\boldsymbol{r}}^T, \boldsymbol{\omega}^T\right)^T$, Eqs. (11) can be rewritten as

$$\begin{gathered} \mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{c}_{\boldsymbol{q}}^T \boldsymbol{\lambda} - \boldsymbol{F}(t, \boldsymbol{q}, \boldsymbol{v}) = \mathbf{0} \\ \boldsymbol{c}(t, \boldsymbol{q}) = \mathbf{0} \end{gathered} \tag{12}$$

where

$$\mathbf{M} = \begin{pmatrix} \mathbf{m} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{t} - \tilde{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} \end{pmatrix}, \quad \boldsymbol{c}_{\boldsymbol{q}} = \begin{pmatrix} \dfrac{\partial \boldsymbol{c}}{\partial \boldsymbol{r}}, \dfrac{\partial \boldsymbol{c}}{\partial \boldsymbol{\pi}} \end{pmatrix}$$

The mass matrix of rigid bodies is constant in the body-fixed coordinate system.

2.2 Dynamics of a Flexible Body with Small Deformations

For a flexible body with small deformation, the FFRF can be used to describe the small elastic deformation. The flexible deformation $\boldsymbol{u}_f$ is described by n coordinates $\boldsymbol{p}$

$$\boldsymbol{u}_f\,(\mathbf{x},\, t) = \mathbf{S}\,(\mathbf{x})\,\boldsymbol{p}(t)$$

where matrix $\mathbf{S} = [\boldsymbol{S}^1, \boldsymbol{S}^2, ..., \boldsymbol{S}^n]$ contains n arbitrary shapes that are required to comply with any kinematic boundary conditions of the problem. The global position of an arbitrary point P on a flexible body is expressed as

$$\boldsymbol{r}_p = \boldsymbol{r}(t) + \mathbf{A}(t)\left(\boldsymbol{u}_0 + \boldsymbol{u}_f\right) = \boldsymbol{r}(t) + \mathbf{A}\bar{\boldsymbol{u}}(t) \tag{13}$$

where $\boldsymbol{u}_0$ is the position of point P in the reference configuration of the body coordinate system. The velocity of point P can be expressed as

$$\dot{\boldsymbol{r}}_p = \dot{\boldsymbol{r}} + \dot{\mathbf{A}}\bar{\boldsymbol{u}} + \mathbf{A}\dot{\bar{\boldsymbol{u}}} = \dot{\boldsymbol{r}} - \mathbf{A}\tilde{\bar{\boldsymbol{u}}}\boldsymbol{\omega} + \mathbf{A}\mathbf{S}\dot{\boldsymbol{p}}$$

The kinetic energy of a flexible body can be written as

$$\begin{aligned}\mathscr{K} &= \frac{1}{2}\int_V \rho\dot{\boldsymbol{r}}_p\cdot\dot{\boldsymbol{r}}_p dv = \frac{1}{2}\dot{\boldsymbol{r}}^T\underbrace{\int_V \rho\mathbf{I}_3 dv}_{\mathbf{m}}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{\omega}^T\underbrace{\int_V \rho\tilde{\bar{\boldsymbol{u}}}^T\tilde{\bar{\boldsymbol{u}}}dv}_{\hat{\mathbf{J}}}\boldsymbol{\omega} + \frac{1}{2}\dot{\boldsymbol{p}}^T\underbrace{\int_V \rho\mathbf{S}^T\mathbf{S}dv}_{\hat{\mathbf{M}}}\dot{\boldsymbol{p}} \\ &\quad - \dot{\boldsymbol{r}}^T\boldsymbol{A}\underbrace{\int_V \rho\tilde{\bar{\boldsymbol{u}}}dv}_{\boldsymbol{\Phi}}\boldsymbol{\omega} + \dot{\boldsymbol{r}}^T\boldsymbol{A}\underbrace{\int_V \rho\mathbf{S}dv}_{\boldsymbol{\Psi}}\dot{\boldsymbol{p}} + \boldsymbol{\omega}^T\underbrace{\int_V \rho\tilde{\bar{\boldsymbol{u}}}\mathbf{S}}_{\boldsymbol{\Gamma}}\dot{\boldsymbol{p}} \\ &= \frac{1}{2}\dot{\boldsymbol{r}}^T\mathbf{m}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{\omega}^T\hat{\mathbf{J}}\boldsymbol{\omega} + \frac{1}{2}\dot{\boldsymbol{p}}^T\hat{\mathbf{M}}\dot{\boldsymbol{p}} - \dot{\boldsymbol{r}}^T\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\omega} + \dot{\boldsymbol{r}}^T\boldsymbol{A}\boldsymbol{\Psi}\dot{\boldsymbol{p}} + \boldsymbol{\omega}^T\boldsymbol{\Gamma}\dot{\boldsymbol{p}}\end{aligned}$$

where the matrices $\mathbf{m}$, $\hat{\mathbf{M}}$, and $\boldsymbol{\Psi}$ are constant, and the matrices including $\mathbf{J}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Gamma}$ depend on the value of the modal coordinates $\boldsymbol{p}$, so that they can be rewritten in a more clear pattern as

$$\begin{aligned}\hat{\mathbf{J}} &= \underbrace{\int_V \rho\tilde{\boldsymbol{u}}_0^T\tilde{\boldsymbol{u}}_0 dv}_{\hat{\mathbf{J}}_0} + p_i\underbrace{\int_V \rho\tilde{\boldsymbol{S}}^{i\,T}\tilde{\boldsymbol{u}}_0 dv}_{\boldsymbol{\Omega}^i} + p_i\int_V \rho\tilde{\boldsymbol{u}}_0^T\tilde{\boldsymbol{S}}^i dv + p_i\underbrace{\int_V \rho\widetilde{\boldsymbol{S}^i}^T\widetilde{\boldsymbol{S}^j}dv p_j}_{\boldsymbol{\Upsilon}^{ij}} \\ &= \mathbf{J}_0 + (\boldsymbol{\Omega}^i + \boldsymbol{\Omega}^{iT})p_i + \boldsymbol{\Upsilon}^{ij}p_i p_j\end{aligned}$$

and

$$\boldsymbol{\Gamma}^i = \int_V \rho\tilde{\bar{\boldsymbol{u}}}\boldsymbol{S}^i dv = \underbrace{\int_V \rho\tilde{\boldsymbol{u}}_0\boldsymbol{S}^i dv}_{\boldsymbol{\gamma}^i} + p_j\underbrace{\int_V \rho\tilde{\boldsymbol{S}}^j\boldsymbol{S}^i dv}_{-\boldsymbol{\eta}^{ij}} = \boldsymbol{\gamma}^i - \boldsymbol{\eta}^{ij}p_j$$

$$\boldsymbol{\Phi} = \int_V \rho\tilde{\boldsymbol{u}}_0 dv + \int_V \rho\widetilde{\mathbf{S}\boldsymbol{p}}dv = \widetilde{m\mathring{\boldsymbol{r}}_c} + \widetilde{\boldsymbol{\Psi}\boldsymbol{p}}$$

where $\mathring{\boldsymbol{r}}_c$ is the center of mass of the body in the reference coordinate system (RCS), and $\mathring{\boldsymbol{r}}_c = \boldsymbol{0}$ when the origin of the RCS coincides with the center of mass of the body.

The strain vector can generally be written as $\boldsymbol{\epsilon} = \mathfrak{D}\mathbf{S}\boldsymbol{p}$, where $\mathfrak{D}$ is the linear spatial differential operator that calculates strains from deformations. The elastic potential energy is

$$\mathscr{U} = \frac{1}{2}\int_V \boldsymbol{\epsilon}^T\mathbf{E}\boldsymbol{\epsilon}dv = \frac{1}{2}\boldsymbol{p}^T\underbrace{\int_V (\mathfrak{D}\mathbf{S})^T\mathbf{E}(\mathfrak{D}\mathbf{S})\,dv}_{\mathbf{K}}\boldsymbol{p} = \frac{1}{2}\boldsymbol{p}^T\mathbf{K}\boldsymbol{p}$$

where $\mathbf{E}$ is the symmetric matrix of the elastic coefficients.

The virtual displacement of point P can be derived from Eq. (13)

$$\delta\boldsymbol{r}_p = \delta\boldsymbol{r} + \delta\mathbf{A}\bar{\boldsymbol{u}} + \mathbf{A}\delta\bar{\boldsymbol{u}} = \delta\boldsymbol{r} + \mathbf{A}\tilde{\bar{\boldsymbol{u}}}^T\delta\boldsymbol{\pi} + \mathbf{A}\mathbf{S}\delta\boldsymbol{p} = \delta\boldsymbol{r} + \mathbf{A}\tilde{\bar{\boldsymbol{u}}}^T\mathbf{G}\delta\boldsymbol{\theta} + \mathbf{A}\mathbf{S}\delta\boldsymbol{p}$$

Hence, the virtual work of an arbitrary distributed force $\mathbf{f}$ applied on the body is expressed by

$$\delta\mathscr{W} = \int_V \delta\boldsymbol{r}_p^T\mathbf{f}dv = \delta\boldsymbol{r}^T\boldsymbol{f}(t) + \delta\boldsymbol{\theta}^T\mathbf{G}^T\boldsymbol{t}(t) + \delta\boldsymbol{q}^T\boldsymbol{m}(t)$$

where

$$\boldsymbol{f}=\int_V \mathbf{f}dv, \quad \mathbf{G}^T\boldsymbol{t}=\int_V \mathbf{G}^T\tilde{\boldsymbol{u}}\mathbf{A}^T\mathbf{f}dv, \quad \boldsymbol{m}=\int_V \mathbf{S}^T\mathbf{A}^T\mathbf{f}dv$$

are the generalized forces with respect to the virtual displacement, $\delta\boldsymbol{r}$, rotation, $\delta\boldsymbol{\theta}$, and modal coordinates, $\delta\boldsymbol{p}$, respectively.

From the Lagrange equations of the first kind, the term with respect to translational coordinates, $\delta\boldsymbol{r}$, is

$$\mathbf{m}\ddot{\boldsymbol{r}}-\mathbf{A}\boldsymbol{\Phi}\dot{\boldsymbol{\omega}}+\mathbf{A}\boldsymbol{\Psi}\ddot{\boldsymbol{p}}-\mathbf{A}\tilde{\boldsymbol{\omega}}\boldsymbol{\Phi}\boldsymbol{\omega}+2\mathbf{A}\tilde{\boldsymbol{\omega}}\boldsymbol{\Psi}\dot{\boldsymbol{p}}+\left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{r}}\right)^T\boldsymbol{\lambda}=\boldsymbol{f} \tag{14}$$

that related to the rotational coordinates, $\delta\boldsymbol{\theta}$, yields

$$\mathbf{G}^T\left\{\hat{\mathbf{J}}\dot{\boldsymbol{\omega}}+\boldsymbol{\Gamma}\ddot{\boldsymbol{p}}-\boldsymbol{\Phi}^T\mathbf{A}^T\ddot{\boldsymbol{r}}+\tilde{\boldsymbol{\omega}}\hat{\mathbf{J}}\boldsymbol{\omega}+2\dot{p}_i(\boldsymbol{\Omega}_i+\boldsymbol{\Upsilon}^{ji}p_i)\boldsymbol{\omega}\right\}+\left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{\theta}}\right)^T\boldsymbol{\lambda}=\mathbf{G}^T\boldsymbol{t} \tag{15}$$

and that related to the modal coordinates $\delta\boldsymbol{p}$, yields

$$\hat{\mathbf{M}}\ddot{\boldsymbol{p}}+\boldsymbol{\Psi}^T\mathbf{A}^T\ddot{\boldsymbol{r}}+\boldsymbol{\Gamma}^T\dot{\boldsymbol{\omega}}-2\boldsymbol{\omega}^T\boldsymbol{\eta}^{ij}\dot{p}_j-\boldsymbol{\omega}^T(\boldsymbol{\Omega}_i+\boldsymbol{\Upsilon}^{ij}p_i)\boldsymbol{\omega}+\left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{p}}\right)^T\boldsymbol{\lambda}=\boldsymbol{m} \tag{16}$$

Similarly, replacing $\dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\theta}}=\dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}\mathbf{G}$ in Eq. (15) and collecting matrix $\mathbf{G}^T$ yield

$$\mathbf{G}^T\left\{\hat{\mathbf{J}}\dot{\boldsymbol{\omega}}+\boldsymbol{\Gamma}\ddot{\boldsymbol{p}}-\boldsymbol{\Phi}^T\mathbf{A}^T\ddot{\boldsymbol{r}}+\tilde{\boldsymbol{\omega}}\hat{\mathbf{J}}\boldsymbol{\omega}+2\dot{p}_i\left(\boldsymbol{\Omega}_i+\boldsymbol{\Upsilon}^{ji}p_i\right)\boldsymbol{\omega}+\left(\frac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}\right)^T\boldsymbol{\lambda}\right\}=\mathbf{G}^T\boldsymbol{t} \tag{17}$$

Recalling that matrix $\mathbf{G}$ should not become singular thanks to the use of relative rotations, premultiplication by matrix $\mathbf{G}^T$ can be eliminated from both sides of the equation without affecting the solution, turning Eq. (17) into a true moment equilibrium equation. From Eqs. (14), (16), and (17), the governing equations can be collected as in [43], yielding

$$\begin{gathered}\mathbf{M}(\boldsymbol{q})\dot{\boldsymbol{v}}+\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{\lambda}-\boldsymbol{F}(t,\boldsymbol{q},\boldsymbol{v})=\mathbf{0}\\ \boldsymbol{c}(t,\boldsymbol{q})=\mathbf{0}\end{gathered} \tag{18}$$

where $\boldsymbol{q}=\left(\boldsymbol{r}^T,\boldsymbol{\theta}^T,\boldsymbol{p}^T\right)^T$, $\boldsymbol{v}=\left(\dot{\boldsymbol{r}}^T,\boldsymbol{\omega}^T,\dot{\boldsymbol{p}}^T\right)^T$,

$$\mathbf{M}=\begin{pmatrix}\mathbf{m} & -\mathbf{A}\boldsymbol{\Phi} & \mathbf{A}\boldsymbol{\Psi}\\ -\boldsymbol{\Phi}^T\mathbf{A}^T & \hat{\mathbf{J}} & \boldsymbol{\Gamma}\\ \boldsymbol{\Psi}^T\mathbf{A}^T & \boldsymbol{\Gamma}^T & \hat{\mathbf{M}}\end{pmatrix}, \quad \boldsymbol{c}_{\boldsymbol{q}}=\begin{pmatrix}\dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{r}}, & \dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{\pi}}, & \dfrac{\partial\boldsymbol{c}}{\partial\boldsymbol{p}}\end{pmatrix}$$

and

$$\boldsymbol{F}=\begin{pmatrix}\boldsymbol{f}-2\mathbf{A}\tilde{\boldsymbol{\omega}}\boldsymbol{\Psi}\dot{\boldsymbol{p}}+\mathbf{A}\tilde{\boldsymbol{\omega}}\boldsymbol{\Phi}\boldsymbol{\omega}\\ \boldsymbol{t}-\tilde{\boldsymbol{\omega}}\hat{\mathbf{J}}\boldsymbol{\omega}-2\dot{p}_i\left(\boldsymbol{\Omega}_i+\boldsymbol{\Upsilon}^{ji}p_j\right)\boldsymbol{\omega}\\ \boldsymbol{m}+2\boldsymbol{\omega}^T\boldsymbol{\eta}^{ij}\dot{p}_j+\boldsymbol{\omega}^T\left(\boldsymbol{\Omega}_i-\boldsymbol{\Upsilon}^{ij}p_i\right)\boldsymbol{\omega}-\mathbf{K}\boldsymbol{p}\end{pmatrix}$$

The derived Eq. (18) is equivalent to the matrix form in [44], where a different process of derivation is provided. It should be noticed that the mass matrix $\mathbf{M}$ is a function of the generalized coordinates, and theoretically, it needs to be evaluated at each time step during the simulation.

2.3 Dynamics of a Flexible Beam with Large Deformations

Suppose that the undeformed length of the beam is L; at time t, the line of the centroids is expressed as $\boldsymbol{r}(x,t)$, where $x \in [0,L]$ is the arc-length coordinate of that line for the undeformed body. The cross-section fiber can be described by two directional vectors in the cross-section plane $\boldsymbol{r}_y(x,t)$ and $\boldsymbol{r}_z(x,t)$. As shown in Fig. 1, the global position of an arbitrary point P, labeled by its material coordinates (x,y,z) on the section, can be written as

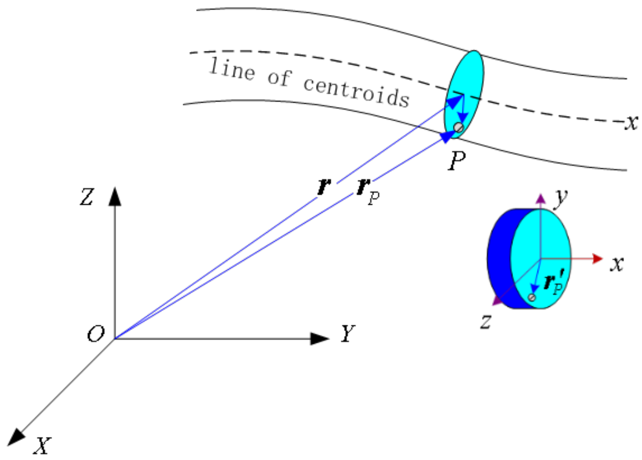


Figure 1: Schematic of a flexible rod

$$\boldsymbol{r}_p(x,y,z,t) = \boldsymbol{r}(x,t) + \boldsymbol{r}'_p(y,z,t) = \boldsymbol{r}(x,t) + y\boldsymbol{r}_y(x,t) + z\boldsymbol{r}_z(x,t)$$

The velocity of an arbitrary point P is

$$\dot{\boldsymbol{r}}_p(x,y,z,t) = \dot{\boldsymbol{r}}(x,t) + y\dot{\boldsymbol{r}}_y(x,t) + z\dot{\boldsymbol{r}}_z(x,t)$$

The kinetic energy of a body can be written as

$$\mathscr{K} = \frac{1}{2}\int_0^L \int_a \rho \dot{\boldsymbol{r}}_p \cdot \dot{\boldsymbol{r}}_p \, da \, dv = \frac{1}{2}\int_0^L \left(\rho_A \dot{\boldsymbol{r}} \cdot \dot{\boldsymbol{r}} + J_{zz}\dot{\boldsymbol{r}}_z \cdot \dot{\boldsymbol{r}}_z + J_{yy}\dot{\boldsymbol{r}}_y \cdot \dot{\boldsymbol{r}}_y - 2J_{yz}\dot{\boldsymbol{r}}_y \cdot \dot{\boldsymbol{r}}_z\right) dx \tag{19}$$

where

$$\rho_A = \int_a \rho(x) da, \quad J_{zz} = \int_a \rho(x) z^2 da, \quad J_{yy} = \int_a \rho(x) y^2 da, \quad J_{yz} = -\int_a \rho(x) yz da$$

Therefore, for a flexible body with N elements, the kinetic energy of each nth element $(n = 1,2,\cdots,N)$ can be expressed by Eq. (19) with its generalized coordinates $\boldsymbol{q}_\alpha$ at the αth node $(\alpha = 1,\cdots,A)$ on the element. The vector $\boldsymbol{q}_\alpha(x,t)$ is constituted by the position vector $\boldsymbol{r}(x_\alpha,t)$ and its gradients $\boldsymbol{r}_x(x_\alpha,t)$, $\boldsymbol{r}_y(x_\alpha,t)$, and $\boldsymbol{r}_z(x_\alpha,t)$ at each node. The vector fields $\boldsymbol{r}(x,t)$, $\boldsymbol{r}_y(x,t)$, and $\boldsymbol{r}_z(x,t)$ in the nth element are interpolated as

$$\boldsymbol{r}(x,t) = s^{\alpha}(x)\boldsymbol{q}_{\alpha}(t), \quad \boldsymbol{r}_y(x,t) = g^{\alpha}(x)\boldsymbol{q}_{\alpha}(t), \quad \boldsymbol{r}_z(x,t) = h^{\alpha}(x)\boldsymbol{q}_{\alpha}(t) \tag{20}$$

where $s^{\alpha}(x)$, $g^{\alpha}(x)$, and $h^{\alpha}(x)$ are the shape functions. Therefore, substituting Eqs. (20) in Eq. (19), the kinetic energy of the nth element is expressed as

$$\mathscr{K} = \frac{1}{2} m_n^{\alpha\kappa} \dot{\boldsymbol{q}}_{\alpha}^T(t)\dot{\boldsymbol{q}}_{\kappa}(t), \quad \kappa = 1,2,\cdots,A \tag{21}$$

where

$$m_n^{\alpha\kappa} = \int_{x_{n-1}}^{x_n} \left[\rho_A s^{\alpha} s^{\kappa} + J_{zz} g^{\alpha} g^{\kappa} - J_{yz}(g^{\alpha} h^{\kappa} + h^{\alpha} g^{\kappa} + J_{yy} h^{\alpha} h^{\kappa})\right] dx$$

The parameters $m_n^{\alpha\kappa}$ are constant and thus can be computed beforehand, and the mass matrix of nth element is $\mathbf{M}_n^e = m_n^{\alpha\beta}\mathbf{I}_3$.

For the potential energy, suppose the reference configuration of the body is $\mathring{\boldsymbol{r}}(x)$, and on the central line we denote

$$\bar{\mathbf{G}}(x,t) = \frac{\partial \boldsymbol{r}(x,t)}{\partial \boldsymbol{x}}, \quad \mathring{\bar{\mathbf{G}}}(x) = \frac{\partial \mathring{\boldsymbol{r}}(x)}{\partial \boldsymbol{x}}$$

Then the strain tensor on the line of centroids can be written as

$$\bar{\boldsymbol{\epsilon}}(x,t) = \frac{1}{2}\mathring{\bar{\mathbf{G}}}^{-T}(x)\left(\bar{\mathbf{G}}^T(x,t)\bar{\mathbf{G}}(x,t) - \mathring{\bar{\mathbf{G}}}^T(x)\mathring{\bar{\mathbf{G}}}(x)\right)\mathring{\bar{\mathbf{G}}}^{-1}(x)$$

Denote

$$\bar{\boldsymbol{k}}_i(x,t) = \bar{\mathbf{G}}^{-1}(x,t)\cdot \boldsymbol{r}_{xi}(x,t), \quad i = y,z$$

For an arbitrary point on the body, the direct computation yields

$$\mathbf{G}(\boldsymbol{x},t) = \left[\boldsymbol{r}_x + y\boldsymbol{r}_{yx} + z\boldsymbol{r}_{zx}, \boldsymbol{r}_y, \boldsymbol{r}_z\right] = \bar{\mathbf{G}}(x,t)\left[\mathbf{I} + (y\bar{\boldsymbol{k}}_y(x,t) + z\bar{\boldsymbol{k}}_z(x,t))\boldsymbol{e}_1^T\right] \tag{22}$$

where $\boldsymbol{e}_1 = (1,0,0)^T$. On the reference configuration, it yields

$$\mathring{\mathbf{G}}(\boldsymbol{x}) = \left[\mathring{\boldsymbol{r}}_x + y\mathring{\boldsymbol{r}}_{yx} + z\mathring{\boldsymbol{r}}_{zx}, \mathring{\boldsymbol{r}}_y, \mathring{\boldsymbol{r}}_z\right] = \mathring{\bar{\mathbf{G}}}(x,t)\left[\mathbf{I} + (y\mathring{\bar{\boldsymbol{k}}}_y(x,t) + z\mathring{\bar{\boldsymbol{k}}}_z(x,t))\boldsymbol{e}_1^T\right] \tag{23}$$

Therefore, from the Eq. (22) and Eq. (23), the Cauchy-Green strain tensor can be written as

$$\boldsymbol{\epsilon}(\boldsymbol{x},t) \approx \bar{\boldsymbol{\epsilon}}(x,t) + y\bar{\mathbf{K}}^y(x,t) + z\bar{\mathbf{K}}^z(x,t) \tag{24}$$

where

$$\bar{\mathbf{K}}^i(x,t) = \frac{1}{2}\mathring{\bar{\mathbf{G}}}^{-T}(x,t)\left[\boldsymbol{k}_i(x,t)\boldsymbol{e}_1^T + \boldsymbol{e}_1\boldsymbol{k}_i^T(x,t)\right]\mathring{\bar{\mathbf{G}}}^{-1}(x,t), \quad i = y,z$$

with the following curvature tensors

$$\boldsymbol{k}_i(x,t) = \bar{\mathbf{G}}^T(x,t)\cdot \boldsymbol{r}_{xi}(x,t) - \mathring{\bar{\mathbf{G}}}^T(x)\cdot \mathring{\boldsymbol{r}}_{xi}(x), \quad i = y,z$$

After obtaining the Cauchy-Green strain tensor $\boldsymbol{\epsilon}(\boldsymbol{x}, t)$, the potential energy can be obtained if the stiffness tensor is known. By discretizing the potential energy using Eq. (20) and computing its virtual perturbation, the element elastic force $\boldsymbol{F}_e^{\alpha}$ can be obtained. More details are reported in [61].

The virtual displacement of an arbitrary point P on the body can be expressed by

$$\delta \boldsymbol{r}_p(x, y, z, t) = \delta \boldsymbol{r}(x, t) + y\delta \boldsymbol{r}_y(x, t) + z\delta \boldsymbol{r}_z(x, t) = \left(s^{\alpha}(x) + yg^{\alpha}(x) + zh^{\alpha}(x)\right)\delta \boldsymbol{q}_{\alpha}(t)$$

Suppose the distributed force $\mathbf{f}$ is applied to the deformable body, then its corresponding generalized force on nth element is

$$\boldsymbol{Q}_n^{\alpha} = \int_{x_{n-1}}^{x_n} \left[s^{\alpha}(x) \int_a \mathbf{f} da + g^{\alpha}(x) \int_a y\mathbf{f} da + h^{\alpha}(x) \int_a z\mathbf{f} da \right] dx \tag{25}$$

As a result, the generalized coordinates $\boldsymbol{x}$ of a body are the collection of the generalized coordinates of all elements and can be expressed as $\boldsymbol{x} = [\boldsymbol{q}_1^T, \cdots, \boldsymbol{q}_{\alpha}^T, \cdots]^T$. The mass matrix $\mathbf{M}$ can be assembled from the element matrix $\mathbf{M}_n^e$. The generalized force $\boldsymbol{F}$ of a body can be assembled from the elastic force $\boldsymbol{F}_e^{\alpha}$ and generalized force $\boldsymbol{Q}_n^{\alpha}$. Therefore, the equations of motion of the constrained dynamical system can be written as

$$\begin{gathered} \mathbf{M}\ddot{\boldsymbol{x}} + \boldsymbol{c}_{\boldsymbol{x}}^T \boldsymbol{\lambda} - \boldsymbol{F}(t, \boldsymbol{x}, \dot{\boldsymbol{x}}) = \mathbf{0} \\ \boldsymbol{c}(t, \boldsymbol{x}) = \mathbf{0} \end{gathered} \tag{26}$$

where $\boldsymbol{c}_{\boldsymbol{x}} = \dfrac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}$ and matrix $\mathbf{M}$ is constant.

2.4 Constrained System Dynamics with Index Reduction

From Eqs. (12), (18), and (26), the index-3 governing equations of rigid and flexible bodies subjected to either small or large deformation can be all written in the form

$$\begin{gathered} \mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{c}_{\boldsymbol{q}}^T(t, \boldsymbol{q})\boldsymbol{\lambda} - \boldsymbol{F}(t, \boldsymbol{q}, \boldsymbol{v}) = \mathbf{0} \\ \boldsymbol{c}(t, \boldsymbol{q}) = \mathbf{0} \end{gathered} \tag{27}$$

Equation (9) shows that the angular velocity $\boldsymbol{\omega}$ and the relative rotation vector $\boldsymbol{\theta}$ are related through matrix $\mathbf{G}$. Therefore, in rigid and flexible systems, the relationship between the generalized velocities and generalized coordinates can be expressed by

$$\boldsymbol{v} = \mathbf{U}\dot{\boldsymbol{q}} \tag{28}$$

where matrix $\mathbf{U}$ is defined in Table 1, which also summarizes the specific form and the generic expressions that Eqs. (27) assume in each case.

The index of the DAE problem is reduced by simultaneously enforcing constraint equations at the position and velocity levels, which requires the introduction of an additional set of Lagrange multipliers, $\boldsymbol{\mu}$, which yields the well-known stabilization

Table 1: Summary of different formulations

	$\mathbf{M}$	$\boldsymbol{q}$	$\boldsymbol{v}$	$\boldsymbol{c}_{\boldsymbol{q}}$	$\mathbf{U}$
Rigid	const	$\left[\ \boldsymbol{r}^T, \boldsymbol{\theta}^T\ \right]^T$	$\left[\ \dot{\boldsymbol{r}}^T, \boldsymbol{\omega}^T\ \right]^T$	$\left[\ \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{r}}, \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\pi}}\ \right]$	$\begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}$
FFRF	non-const	$\left[\ \boldsymbol{r}^T, \boldsymbol{\theta}^T, \boldsymbol{p}^T\ \right]^T$	$\left[\ \dot{\boldsymbol{r}}^T, \boldsymbol{\omega}^T, \dot{\boldsymbol{p}}^T\ \right]^T$	$\left[\ \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{r}}, \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\pi}}, \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{p}}\ \right]$	$\begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_n \end{bmatrix}$
ANCF	const	$\boldsymbol{x}$	$\dot{\boldsymbol{x}}$	$\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}$	$\mathbf{I}_m$

method proposed in [42]. The following stabilized index-2 DAE system is obtained,

$$\mathbf{M}\dot{\boldsymbol{v}} + \boldsymbol{c}_{\boldsymbol{q}}^T\left(t, \boldsymbol{q}\right)\boldsymbol{\lambda} - \boldsymbol{F}\left(t, \boldsymbol{q}, \boldsymbol{v}\right) = \mathbf{0} \tag{29a}$$

$$\boldsymbol{c}_{\boldsymbol{q}}\left(t, \boldsymbol{q}\right)\boldsymbol{v} + \boldsymbol{c}_t\left(t, \boldsymbol{q}\right) = \mathbf{0} \tag{29b}$$

$$\mathbf{M}\left(\mathbf{U}\dot{\boldsymbol{q}} - \boldsymbol{v}\right) + \boldsymbol{c}_{\boldsymbol{q}}^T\left(t, \boldsymbol{q}\right)\boldsymbol{\mu} = \mathbf{0} \tag{29c}$$

$$\boldsymbol{c}\left(t, \boldsymbol{q}\right) = \mathbf{0} \tag{29d}$$

where $\boldsymbol{\mu}$ can be interpreted as a projection term in the kinematical differential equation. For planar problems, one rotational parameter is capable to describe the rotational motion, yielding $\mathbf{U} = \mathbf{I}$.

3 Constraint Stabilization

To derive the underlying ODEs, two methods, i.e., the algebraic elimination of the Lagrange multipliers and the coordinate partitioning method, are practicable. The first method eliminates the Lagrange multipliers to obtain a set of ODEs, including Baumgarte's stabilization technique [46], the augmented Lagrangian formulation by Bayo and Ledesman [47], a self-stabilized algorithm [48], and Gear's penalty methods [49]. The latter method eliminates the constraint equation by selecting a minimal set of coordinates; a practical coordinate partitioning approach can be seen in [50]. Without exception, all these methods underline the difficulty of dealing with constraint drift. A detailed review of constraint enforcement methods can be found in [51].

In this paper, a stabilization technique is adopted to obtain a set of ODEs by eliminating the Lagrange multipliers. An algebraic constraint equation

$$f(t, \boldsymbol{x}) = 0 \tag{30}$$

is replaced by a corresponding first order differential equation

$$\dot{f} + \beta f = f_{\boldsymbol{x}}(t, \boldsymbol{x})\dot{\boldsymbol{x}} + f_t(t, \boldsymbol{x}) + \beta f(t, \boldsymbol{x}) = 0 \tag{31}$$

whose global asymptotically stable solution

$$f(t, \boldsymbol{x}) = f(t_0, \boldsymbol{x})e^{-\beta t} \tag{32}$$

approaches zero, $f = 0$, with exponential rate, when the parameter β is positive. When it is large enough, any constraint drift and error in initial values will be quickly pulled back onto the constraint manifold. A first order Taylor expansion of Eq. (30) yields

$$f(t+h, \boldsymbol{x}+d\boldsymbol{x}) \approx f(t,\boldsymbol{x}) + f_{\boldsymbol{x}}(t,\boldsymbol{x})d\boldsymbol{x} + f_t(t,\boldsymbol{x})h \tag{33}$$

Prescribing the vanishing of the higher-order term $f(t+h, \boldsymbol{x}+d\boldsymbol{x}) \approx 0$, Eq. (33) is equivalent to Eq. (31) with $d\boldsymbol{x} = \dot{\boldsymbol{x}}h$ and $\beta = 1/h$. In this sense, Eq. (31) shows some resemblance with the revised holonomic constraints proposed in [52].

By using this technique, i.e. by combining Eq. (29b) with its derivative according to Eq. (31), one obtains

$$\boldsymbol{c}_{\boldsymbol{q}}\dot{\boldsymbol{v}} + \left(\boldsymbol{c}_{\boldsymbol{q}}\boldsymbol{v}\right)_{\boldsymbol{q}}\boldsymbol{v} + 2\boldsymbol{c}_{\boldsymbol{q}t}\boldsymbol{v} + \boldsymbol{c}_{tt} + \beta\left(\boldsymbol{c}_{\boldsymbol{q}}\boldsymbol{v} + \boldsymbol{c}_t\right) = \mathbf{0} \tag{34}$$

Denoting $\boldsymbol{\gamma} = \left(\boldsymbol{c}_{\boldsymbol{q}}\boldsymbol{v}\right)_{\boldsymbol{q}}\boldsymbol{v} + 2\boldsymbol{c}_{\boldsymbol{q}t}\boldsymbol{v} + \boldsymbol{c}_{tt}$, extracting $\dot{\boldsymbol{v}}$ from Eq. (29a) and replacing it in Eq. (34), then extracting $\boldsymbol{\lambda}$ and replacing it back in Eq. (29a), one obtains

$$\dot{\boldsymbol{v}} = \left(\mathbf{I} - \mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{\Delta}^{-1}\boldsymbol{c}_{\boldsymbol{q}}\right)\mathbf{M}^{-1}\boldsymbol{F} - \mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{\Delta}^{-1}\left(\boldsymbol{\gamma} + \beta\left(\boldsymbol{c}_{\boldsymbol{q}}\boldsymbol{v} + \boldsymbol{c}_t\right)\right) \tag{35}$$

with $\boldsymbol{\Delta} = \boldsymbol{c}_{\boldsymbol{q}}\mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T$. Similarly, applying the same technique to Eq. (29d) yields

$$\boldsymbol{c}_{\boldsymbol{q}}\mathbf{U}\dot{\boldsymbol{q}} + \boldsymbol{c}_t + \beta\boldsymbol{c} = \mathbf{0} \tag{36}$$

Extracting $\dot{\boldsymbol{q}}$ from Eq. (29c) and replacing it in Eq. (36), then extracting $\boldsymbol{\mu}$ and replacing it back in Eq. (29c), one obtains

$$\dot{\boldsymbol{q}} = \mathbf{U}^{-1}\left\{\left(\mathbf{I} - \mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{\Delta}^{-1}\boldsymbol{c}_{\boldsymbol{q}}\right)\boldsymbol{v} - \mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T\boldsymbol{\Delta}^{-1}\left(\boldsymbol{c}_t + \beta\boldsymbol{c}\right)\right\} \tag{37}$$

The problem is reformulated as the integration of two sets of ODEs, namely Eqs. (35) and (37); the latter also contains the constraint equation, $\boldsymbol{c}$, and the former its first derivative, $\dot{\boldsymbol{c}}$, both multiplied by the relaxation parameter β. To eliminate numerical drift from the constraint manifold, β should be selected depending on the characteristics of the problem. For a non-stiff system, a small value compared to $1/h$ can be chosen, and the ODEs can be solved by explicit integration; implicit ODE integrators are better suited for stiff ODEs, along with larger values for β. If explicit integration is to be used, some specific explicit integration schemes, such as the Runge-Kutta-Chebyshev family of integrators [53], could be a valid choice to deal with mildly stiff subsystems, due to their long-tail stable domain on the negative real axis in the complex plane. Moreover, the solution of the problem can also consist in the integration of Eq. (35), if the Lie group generalized-α ODE integrator [59] is used. In this respect, the computation of the inverse matrix $\mathbf{U}^{-1}$ during each time step is completely avoided.

4 Co-Simulation Process

For a constrained rigid or flexible multibody system, the generalized coordinates of the bodies are partitioned into different subsystems, regardless of being rigid or flexible. Denote that $\boldsymbol{q}_i$ and $\boldsymbol{v}_i$ are the generalized coordinates and velocities of subsystem i. The subsystems are connected by the constraint equations $\boldsymbol{c}(t, \boldsymbol{q}_1, \boldsymbol{v}_1, \boldsymbol{q}_2, \boldsymbol{v}_2 \ldots)$, thus their interaction relies on the exchange of the information $\{\boldsymbol{q}_1, \boldsymbol{v}_1, \boldsymbol{q}_2, \boldsymbol{v}_2 \ldots\}$.

Consider two subsystems, 1 and 2; the implementation of the co-simulation process is given by

$$\dot{\boldsymbol{q}}_1^{k+1} = \boldsymbol{Q}_1\left(t, \boldsymbol{q}_1^{k+1}, \boldsymbol{v}_1^{k+1}, \boldsymbol{q}_2^k, \boldsymbol{v}_2^k\right), \qquad \dot{\boldsymbol{v}}_1^{k+1} = \boldsymbol{V}_1\left(t, \boldsymbol{q}_1^{k+1}, \boldsymbol{v}_1^{k+1}, \boldsymbol{q}_2^k, \boldsymbol{v}_2^k\right) \tag{38a}$$

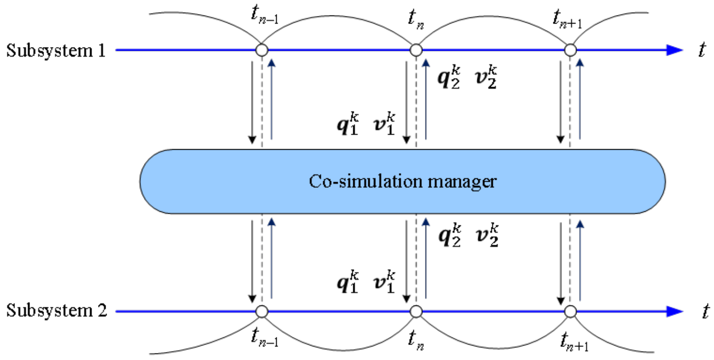
$$\dot{\boldsymbol{q}}_2^{k+1} = \boldsymbol{Q}_2\left(t, \boldsymbol{q}_2^{k+1}, \boldsymbol{v}_2^{k+1}, \boldsymbol{q}_1^k, \boldsymbol{v}_1^k\right), \qquad \dot{\boldsymbol{v}}_2^{k+1} = \boldsymbol{V}_2\left(t, \boldsymbol{q}_2^{k+1}, \boldsymbol{v}_2^{k+1}, \boldsymbol{q}_1^k, \boldsymbol{v}_1^k\right) \tag{38b}$$

where k is the iteration counter, and the functions $\boldsymbol{Q}_i$ and $\boldsymbol{V}_i$ $(i = 1, 2)$ represent the right hand sides of Eqs. (37) and (35) corresponding to the generalized coordinates $\boldsymbol{q}_i$

$$\boldsymbol{Q}_i = \mathbf{U}_i^{-1}\left\{\left(\mathbf{I}_i - \mathbf{M}_i^{-1}\left[\boldsymbol{c}_{\boldsymbol{q}}^T \boldsymbol{\Delta}^{-1} \boldsymbol{c}_{\boldsymbol{q}}\right]_i\right)\boldsymbol{v}^i - \mathbf{M}_i^{-1}\left[\boldsymbol{c}_{\boldsymbol{q}}^T \boldsymbol{\Delta}^{-1}\left(\boldsymbol{c}_t + \beta \boldsymbol{c}\right)\right]_i\right\} \tag{39a}$$

$$\boldsymbol{V}_i = \left(\mathbf{I}_i - \mathbf{M}_i^{-1}\left[\boldsymbol{c}_{\boldsymbol{q}}^T \boldsymbol{\Delta}^{-1} \boldsymbol{c}_{\boldsymbol{q}}\right]_i\right)\mathbf{M}_i^{-1}\boldsymbol{F}_i - \mathbf{M}_i^{-1}\left[\boldsymbol{c}_{\boldsymbol{q}}^T \boldsymbol{\Delta}^{-1}\left(\boldsymbol{\gamma} + \beta\left(\boldsymbol{c}_{\boldsymbol{q}}\boldsymbol{v} + \boldsymbol{c}_t\right)\right)\right]_i \tag{39b}$$

where $\mathbf{M}_i$ is the mass matrix of subsystem i; $\boldsymbol{F}_i$ denotes the generalized force of subsystem i; $\mathbf{U}_i$ is a matrix which is a function of the generalized coordinates of subsystem i, defined according to the formulas in Table 1; $\boldsymbol{\Delta} = \boldsymbol{c}_{\boldsymbol{q}}\mathbf{M}^{-1}\boldsymbol{c}_{\boldsymbol{q}}^T$; the expression $[\cdot]$ denotes the part of the matrix or the vector in the square brackets corresponding to the index of the generalized coordinates of subsystem i; $\mathbf{I}_i$ is the corresponding unitary matrix. Equation (38a) and (38b) represent the equations of subsystems 1 and 2, respectively. Figure 2 depicts the co-simulation implementation of these two sets of equations. As shown in Fig. 2, $\boldsymbol{q}_2$ and $\boldsymbol{v}_2$ in subsystem 1 are obtained from subsystem 2 at communication time nodes, and vice versa. Therefore, in each subsystem, the values of $\mathbf{Q}$ and $\mathbf{V}$ in Eqs. (38a) and (38b) can be evaluated. During a given iteration, ODE integrators are used to solve the Eqs. (38a) and (38b) independently. The information exchanges between subsystems; then, numerical integration is performed again for each subsystem. The above process is repeated until the



t_{n-1}
t_n
t_{n+1}
Subsystem 1
t
$\boldsymbol{q}_2^k$ $\boldsymbol{v}_2^k$
$\boldsymbol{q}_1^k$ $\boldsymbol{v}_1^k$
Co-simulation manager
$\boldsymbol{q}_2^k$ $\boldsymbol{v}_2^k$
$\boldsymbol{q}_1^k$ $\boldsymbol{v}_1^k$
Subsystem 2
t
t_{n-1}
t_n
t_{n+1}

Figure 2: Schematic of the co-simulation process

iteration reaches convergence. During the simulation, the time step size h_i and the stabilization parameter β_i satisfy $\beta_i h_i = 1$. Furthermore, as mentioned in Section 3, the computation of the inverse matrix $\mathbf{U}^{-1}$ can be completely avoided when the Lie group generalized-α ODE integrator is used to solve Eqs. (38).

5 Numerical Examples

The proposed co-simulation scheme is applied to a spatial slider-crank mechanism. The problem is derived from a rigid multibody benchmark[1] proposed by IFToMM's Technical Committee for Multibody Dynamics (`https://www.iftomm-multibody.org/`). It was previously analyzed in [54]. The problem is sketched in Fig. 3. The mechanism consists of a rigid crank AB of length 0.08 m, a connecting rod BC of length 0.3 m, and a rigid sliding block. The crank, connected to the ground by revolute joint A, can rotate freely from the initial position, corresponding to an angle $\theta = 0$ rad, with an initial angular velocity of 6 rad/s. The block is constrained to the ground by a translational joint D that allows sliding along the x axis. A spherical joint at B and a universal joint at C connect the link to the crank and to the slider, respectively. A uniform gravity field of magnitude 9.81 m/s^2 is assumed in the negative z direction. No other load is applied to the system.

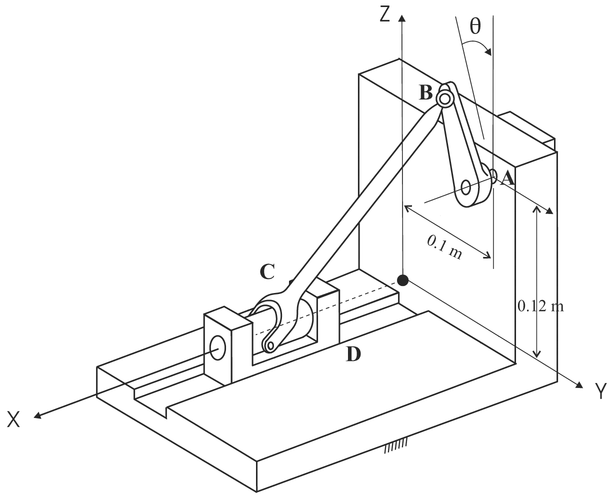


Figure 3: Spatial slider-crank mechanism

[1] See `https://www.iftomm-multibody.org/benchmark/problem/Spatial_rigid_slider-crank_mechanism/` for further details.

The masses of crank and sliding block are $m_c = 0.12$ kg and $m_s = 2.0$ kg, respectively. The mass moments of inertia of the two bodies are

$$\mathbf{J}_c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot 10^{-4}, \qquad \mathbf{J}_s = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot 10^{-4}$$

The center of mass of both crank and rod is assumed to be at their mid-span.

In this work, the connecting rod BC is modeled either as a rigid or flexible body. In the rigid case, to compare with the results originally proposed by Ramin Masoudi, the mass is $m_r = 0.5$ kg, and its mass moments of inertia are

$$\mathbf{J}_r = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 4 \end{pmatrix} \cdot 10^{-3}$$

In the flexible case, the rod is assumed to be uniform, of density $\rho = 7870$ kg/m^3. The cross-section of the rod is assumed to be solid circular. The mass of the rod is $m_r = 0.5$ kg. The total mass moment of inertia slightly differs from the rigid case. In addition, a Young modulus of 207 GPa, and a Poisson ratio of 0.29, are considered. The FFRF and ANCF are considered. For validation purposes, the results of the rigid simulations are compared with those originally proposed by Masoudi on IFToMM's website (where the Rosenbrock method [55] is applied with the relative and absolute tolerances being 1.10^{-4}) and the corresponding results obtained using the free general-purpose multibody solver MBDyn [56] (`http://www.mbdyn.org/`). The results of the flexible models are compared with those from MBDyn.

5.1 Case 1: Rigid Multibody System

When the problem is modeled as a rigid multibody system, the generalized coordinates can be partitioned into two ODE subsystems: the slow subsystem 1 and the fast subsystem 2. Subsystem 1 consists of the generalized coordinates of two rigid bodies: the crank AB (subscript *c*) and the rod BC (subscript *r*),

$$\text{subsystem 1:} \quad \boldsymbol{q}_1 = \left(\boldsymbol{r}_c^T, \boldsymbol{\theta}_c^T, \boldsymbol{r}_r^T, \boldsymbol{\theta}_r^T\right)^T, \quad \boldsymbol{v}_1 = \left(\boldsymbol{v}_c^T, \boldsymbol{\omega}_c^T, \boldsymbol{v}_r^T, \boldsymbol{\omega}_r^T\right)^T$$

Subsystem 2 consists of the slider C (subscript *s*),

$$\text{subsystem 2:} \quad \boldsymbol{q}_2 = \left(\boldsymbol{r}_s^T, \boldsymbol{\theta}_s^T\right)^T, \quad \boldsymbol{v}_2 = \left(\boldsymbol{v}_s^T, \boldsymbol{\omega}_s^T\right)^T$$

The ODE subsystems are formulated according to Eqs. (38). In the co-simulation, to evaluate $\boldsymbol{\Delta}^{-1}$ that appears in functions $\mathbf{Q}$ and $\mathbf{V}$ of Eq. (38), the inverse mass matrices of the mechanism are required. The mass matrices are

$$\mathbf{M}_1 = \operatorname{diag}(\mathbf{m}_c, \mathbf{J}_c, \mathbf{m}_r\, \mathbf{J}_r), \qquad \mathbf{M}_2 = \operatorname{diag}(\mathbf{m}_s, \mathbf{J}_s), \qquad \mathbf{M} = \operatorname{diag}(\mathbf{M}_1, \mathbf{M}_2)$$

where matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are constant. The inverse matrices $\mathbf{M}_1^{-1}$, $\mathbf{M}_2^{-1}$, and $\mathbf{M}^{-1}$ can be calculated beforehand and stored.

As for constraints, the revolute joint at point A provides 5 independent constraint equations, the universal joint at B yields 4 constraint equations, the spherical joint at C provides 3 constraint equations, and the translational joint at D contributes 5 constraint equations. Therefore, the size of the constraint equations vector $\boldsymbol{c}$ is 17×1. Considering the 18 generalized coordinates of the three rigid bodies, the size of $\boldsymbol{c_q}$ is 17×18 and the system has exactly 1 degree of freedom.

The value of the stabilization parameter for both subsystems is chosen as $\beta = 20000$. The size of the time step is chosen as $h = 0.00005$ s, namely $h = 1/\beta$. The Lie group generalized-α ODE integrator with constant step size is used to integrate the ODEs of Eqs. (38), with the Jacobian matrix evaluated numerically. Figure 4 shows that the co-simulation results for position and velocity of the sliding block agree with those obtained analyzing the monolithic problem with MBDyn ($h = 0.004$ s), as well as, for the position alone, with Masoudi's ones as available from IFToMM's website (velocity is not provided). Figure 5 shows that also the co-simulation results for crank angle and angular velocity coincide with the corresponding ones obtained with MBDyn, as well as, for the angle alone, with Masoudi's. The total energy of the rigid system is presented in Fig. 6. The slow decay of the total mechanical energy, despite being the problem conservative, is associated with algorithmic dissipation. In addition, the convergence plot in Fig. 7 presents co-simulation results with different time steps, showing that as the time step size gets smaller the co-simulation results tend to those obtained with the monolithic problem using MBDyn.

5.2 Case 2: Rigid-Flexible Multibody System, with Flexible Connecting Rod

As anticipated, the floating frame of reference and the absolute nodal coordinate formulations are used to describe the deformation of the flexible bodies with the same structural properties provided in an earlier section.

Floating frame of reference formulation

Assuming that the elastic deformation of the rod is small, the floating frame of reference formulation (FFRF) can be used. The generalized coordinates can be partitioned into two ODE subsystems: the slow subsystem 1 and the fast subsystem 2. Subsystem 1 consists of the generalized coordinates of two rigid bodies: the crank AB (subscript c) and the sliding block (subscript s)

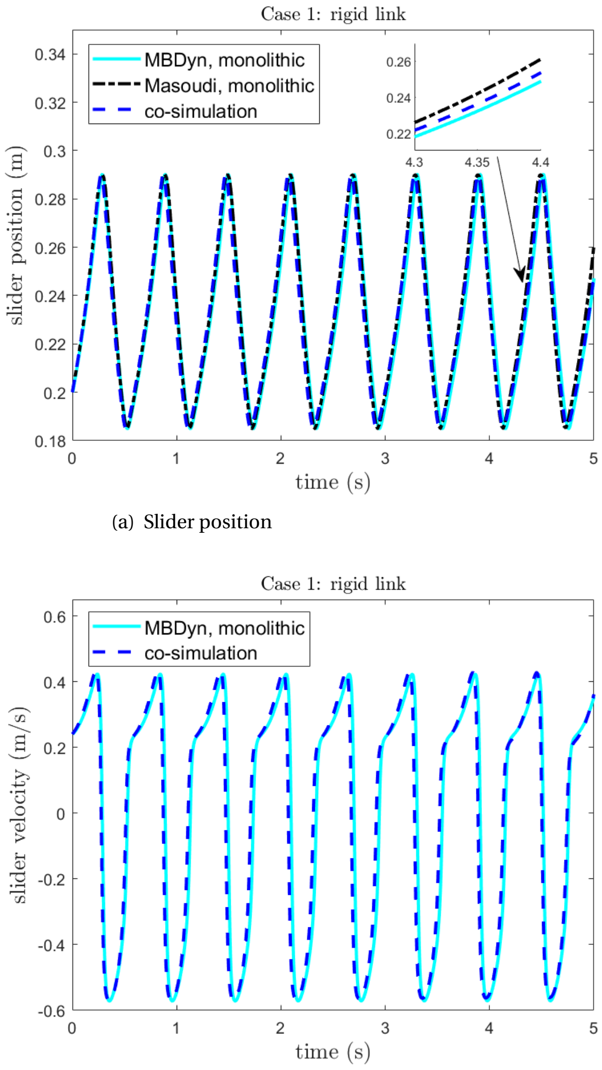
$$\text{subsystem 1:} \quad \boldsymbol{q}_1 = \left(\boldsymbol{r}_c^T, \boldsymbol{\theta}_c^T, \boldsymbol{r}_s^T, \boldsymbol{\theta}_s^T\right)^T, \quad \boldsymbol{v}_1 = \left(\boldsymbol{v}_c^T, \boldsymbol{\omega}_c^T, \boldsymbol{v}_s^T, \boldsymbol{\omega}_s^T\right)^T$$

Subsystem 2 consists of the flexible rod BC (subscript r)

$$\text{subsystem 2:} \quad \boldsymbol{q}_2 = \left(\boldsymbol{r}_r^T, \boldsymbol{\theta}_r^T, \boldsymbol{p}^T\right)^T, \quad \boldsymbol{v}_2 = \left(\boldsymbol{v}_r^T, \boldsymbol{\omega}_r^T, \boldsymbol{u}_{\boldsymbol{p}}^T\right)^T$$

The ODE subsystems are formulated according to Eqs. (38). As discussed in the previous case, to evaluate $\boldsymbol{\Delta}^{-1}$ that appears in the functions $\mathbf{Q}$ and $\mathbf{V}$ of Eq. (38), the inverse mass matrices of the mechanism are required. The mass matrices now are

$$\mathbf{M}_1 = \operatorname{diag}\left(\mathbf{m}_c, \mathbf{J}_c, \mathbf{m}_{s,} \mathbf{J}_s\right), \qquad \mathbf{M}_2 = \bar{\mathbf{M}}, \qquad \mathbf{M} = \operatorname{diag}\left(\mathbf{M}_1, \mathbf{M}_2\right)$$

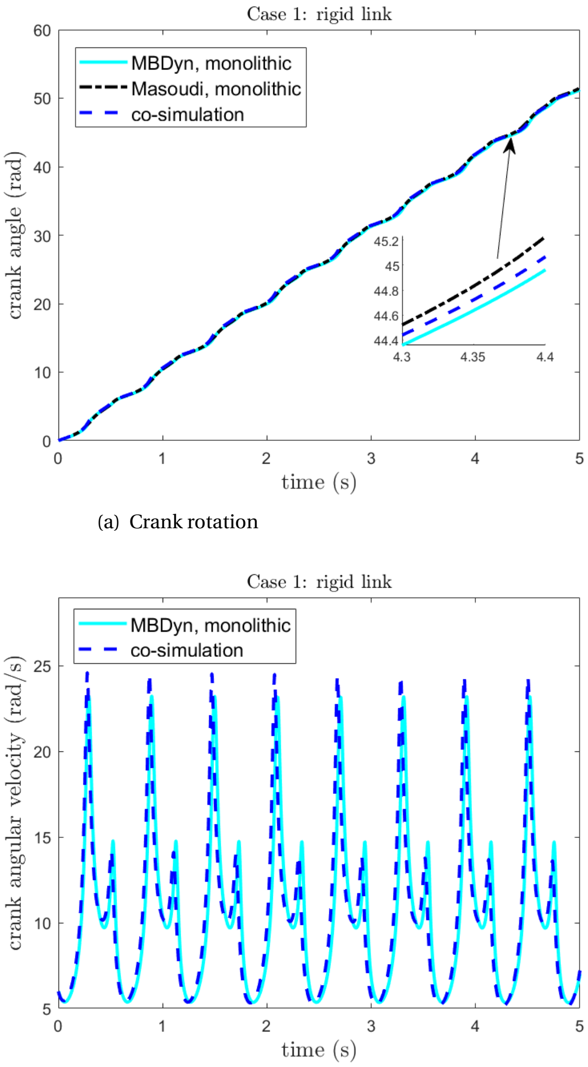


(a) Slider position

(b) Slider velocity

Figure 4: Comparison of slider position and velocity

where $\mathbf{M}_1$ is constant, and $\bar{\mathbf{M}}$ is the mass matrix of the FFRF, which is a matrix function of the modal coordinates. Based on the assumption of small deformations, the modal coordinates can be viewed as minor quantities, whose contribution to the mass matrix can often be neglected. This practice is reasonable because the most significant errors in co-simulation are from delayed data communication, interpolations, and extrapolations. Also in this case, the inverse matrices $\mathbf{M}_1^{-1}$, $\mathbf{M}_2^{-1}$, and $\mathbf{M}^{-1}$ can be calculated beforehand and stored.



(a) Crank rotation

(b) Crank angular velocity

Figure 5: Comparison of crank angle and angular velocity

Absolute nodal coordinate formulation

The generalized coordinates and velocities of the rigid bodies are assumed as the slow variables, while those of the flexible bodies are considered the fast ones. Hence, the generalized coordinates can be partitioned into two ODE subsystems. Subsystem 1 consists of the generalized coordinates of the two rigid bodies: the crank AB

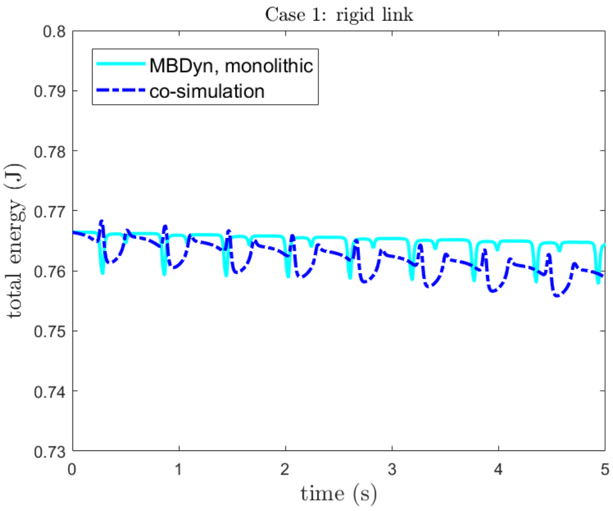


Figure 6: System total energy

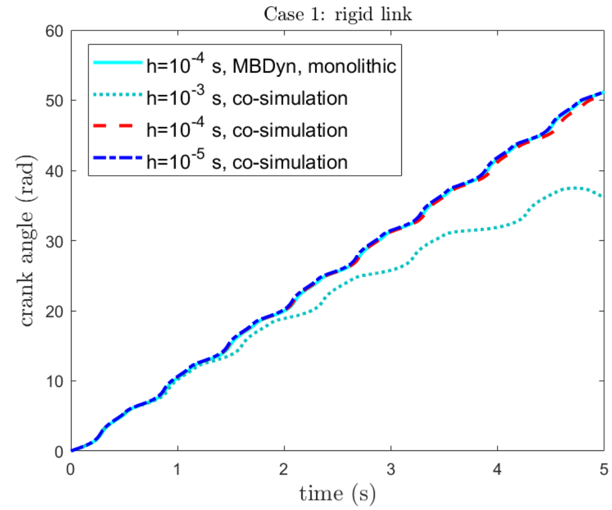


Figure 7: Convergence: comparison of crank angle between monolithic solution by MBDyn and proposed method with various time steps

(subscript c) and the sliding block (subscript s)

$$\text{subsystem 1:} \quad \boldsymbol{q}_1 = \left(\boldsymbol{r}_c^T, \boldsymbol{\theta}_c^T, \boldsymbol{r}_s^T, \boldsymbol{\theta}_s^T\right)^T, \quad \boldsymbol{v}_1 = \left(\boldsymbol{v}_c^T, \boldsymbol{\omega}_c^T, \boldsymbol{v}_s^T, \boldsymbol{\omega}_s^T\right)^T$$

Subsystem 2 consists of the flexible connecting rod BC

$$\text{subsystem 2:} \quad \boldsymbol{q}_2 = \boldsymbol{x}, \quad \boldsymbol{v}_2 = \boldsymbol{u}$$

The two ODE subsystems are formulated according to Eqs. (38). The mass matrices are

$$\mathbf{M}_1 = \mathrm{diag}\left(\mathbf{m}_c, \mathbf{J}_c, \mathbf{m}_s, \mathbf{J}_s\right), \qquad \mathbf{M}_2 = \bar{\mathbf{M}}, \qquad \mathbf{M} = \mathrm{diag}\left(\mathbf{M}_1, \mathbf{M}_2\right)$$

where $\bar{\mathbf{M}}$ is now the mass matrix of the ANCF formulation, and $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}$ are constant matrices. Also in this case the inverse matrices $\mathbf{M}_1^{-1}$, $\mathbf{M}_2^{-1}$, and $\mathbf{M}^{-1}$ can be calculated beforehand and stored.

To describe the deformation of rod BC, the three-dimensional ANCF rod element [43], sketched in Fig. 8, is adopted here. For this element, the position and

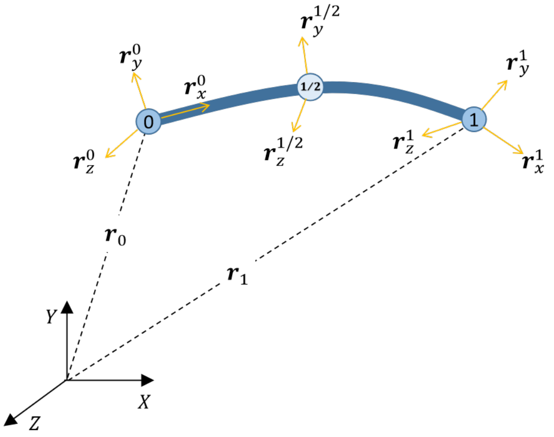


Figure 8: Three-dimensional ANCF rod element

slope vectors at both terminal nodes, as well as the cross-section slope vectors at the middle node constitute the generalized coordinates, amounting to 30 degrees of freedom. The position vector $\boldsymbol{r}(x,t)$ and the slope vectors $\boldsymbol{r}_y(x,t)$ and $\boldsymbol{r}_z(x,t)$ at the nodes are interpolated independently in a way that the position vector at any point along the element is expressed by

$$\boldsymbol{r}(x,t) = (1-3\xi^2+2\xi^3)\boldsymbol{r}_0(t) + l(\xi-2\xi^2+\xi^3)\boldsymbol{r}_x^0(t) + (3\xi^2-2\xi^3)\boldsymbol{r}_1(t) + l(\xi^3-\xi^2)\boldsymbol{r}_x^1(t)$$

where l is the element length and $\xi = x/l$. The cross-section slope vectors at any point on the element can be written as

$$\boldsymbol{r}_\kappa(x,t) = (1-3\xi+2\xi^2)\boldsymbol{r}_\kappa^0(t) + 4\xi(1-\xi)\boldsymbol{r}_\kappa^{1/2}(t) + \xi(2\xi-1)\boldsymbol{r}_\kappa^1(t)$$

where $\kappa = y, z$.

Simulation results

In the FFRF, 6 modal coordinates, namely the first two transverse, the first two lateral, and the first longitudinal and torsional modal shapes, are adopted to describe the deformation. Considering the 12 generalized coordinates of the two rigid bodies and 6 generalized coordinates for the reference motion of the flexible rod, there are

24 generalized coordinates in total. Since the size the of constraint equations vector is 17, matrix $\boldsymbol{c_q}$ is 17×24. The stabilization parameter of both subsystems is $\beta = 10^5$. The time step size is $h = 10^{-5}$ s. The Lie group generalized-α ODE integrator with constant step size is used to solve the ODEs of Eqs. (38), with the Jacobian matrix evaluated numerically.

In the ANCF, 4 elements are used to model the rod, so that the generalized coordinates of the rod is 84. In addition to the 12 generalized coordinates of the two rigid bodies, the total number of generalized coordinates is 96. Hence, the size of the matrix $\boldsymbol{c_q}$ is 17×96. The stabilization parameter for both subsystems is taken as $\beta = 2 \times 10^5$. The Lie group generalized-α ODE integrator is employed here with the step size $h = 5 \times 10^{-6}$ s for both ODE subsystems of Eqs. (38). The resulting position of the sliding block is coincident with the results of the corresponding flexible system analyzed using MBDyn ($h = 0.001$ s) with 5 three-node finite volume beam elements [57, 58], as shown in Fig. 9. The rotation of the crank angle is compared in Fig. 10, showing that the co-simulation results of the spatial rotation in the flexible cases coincide with those of the monolithic solution. The total energy of the flexible system is depicted in Fig. 11. In both cases, the slow decay of the total mechanical energy is caused by the algorithmic dissipation introduced by the numerical scheme and the co-simulation procedure. Specifically, the numerical approximations intrinsic in index reduction and co-simulation manifest themselves as a form of algorithmic dissipation that is magnified through its propagation in the strain energy of the problem.

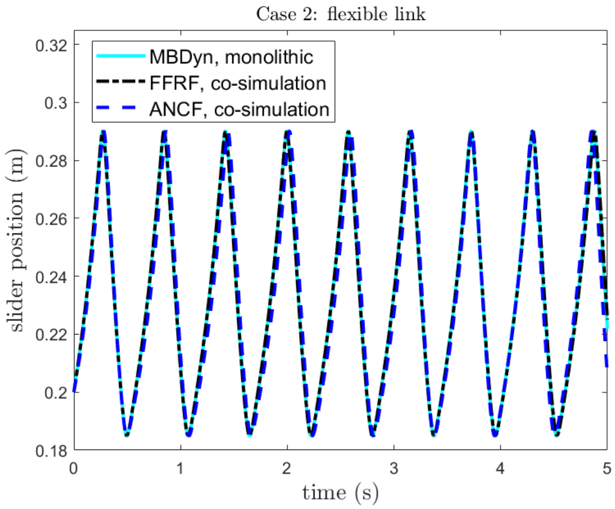


Figure 9: Slider position

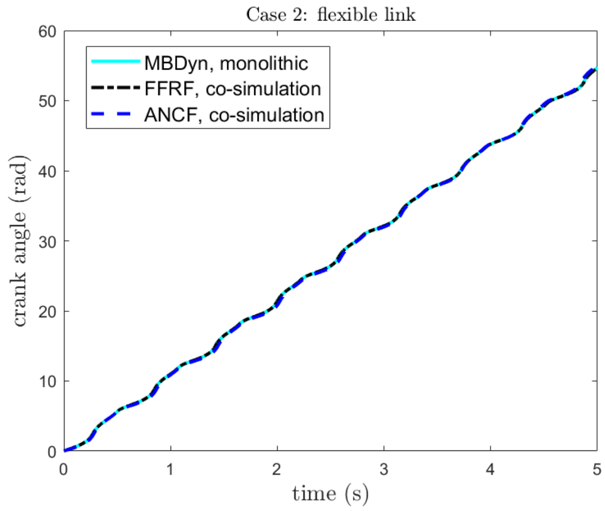


Figure 10: Crank angle

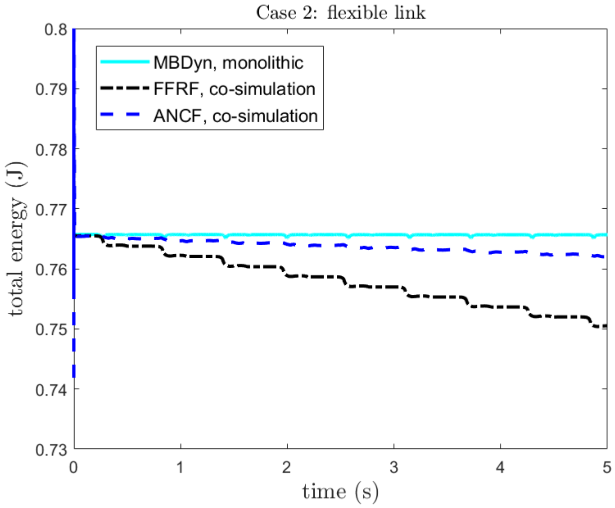


Figure 11: System total energy

5.3 Further Discussion

Constant or block diagonal mass matrices are important features in the simulation of constrained mechanical systems. For example,

1. the mass matrices of rigid bodies and flexible bodies formulated with the ANCF are constant;

2. the mass matrix of a flexible body formulated with the FFRF is a function of the modal coordinates. However, since the values of the modal coordinates are limited, the mass matrix can often be treated as constant, with negligible error compared to that introduced by the decayed data communication. This practice is validated here.

Therefore, the inverse mass matrices in the proposed formulation only need to be computed once and stored for later use.

The simulation of the rigid system is faster than that of flexible ones; among them, the FFRF simulation is much faster than that based on ANCF, thanks to the reduced number of generalized coordinates and its ability to converge when longer time steps are used. The absolute timings are not relevant, as the code was not optimized for performance. For example, considering the rigid system's simulation as the reference, the flexible model based on FFRF requires twice the time, whereas the simulation based on the ANCF is about nine times longer, as detailed in Table 2. This result is attributed to the use of a constant inverse of the mass matrix, the numerical computation of the Jacobian matrix, and the iteration process.

Table 2: Relative computational times

time step	$h = 10^{-4}$	$h = 10^{-5}$
FFRF/rigid (24/18 generalized coordinates)	~2	~2
ANCF/rigid (84/18 generalized coordinates)	(no convergence)	~9

6 Conclusion

A relaxed coupling method is proposed to compute the dynamics of constrained mechanical systems that are connected by joints. Instead of directly trying to solve the index-3 DAEs with a waveform relaxation method, a set of ODEs is obtained and solved iteratively, which makes the implementation straightforward. To achieve this, the constraint equations are enforced by a constraint stabilization technique. Subsystems are formulated using a constant mass matrix, either exactly or through approximation in the floating frame of reference case. Consequently, the inverse of the mass matrix can be stored beforehand without computing it at each time step. The presented method is suitable for the parallel solution of algebraically constrained multibody systems when a Jacobi coupling scheme is used.

7 Acknowledgements

This work is supported by the National Natural Science Foundations of China, with grant number 11772101, the Shanghai Academy of Spaceflight Technology, with grant number SAST 2017-025, the National Key Research and Development Project of China under grant number 2018YFF0300506, and 2020 China Scholarship Council.

References

[1] Gomes, C., Thule, C., Broman, D., Larsen, P. G., Vangheluwe, H.: Co-simulation: state of the art (2017). https://arxiv.org/abs/1702.00686

[2] Eberhard, P.: IUTAM Symposium on multiscale problems in multibody system contacts. Proceedings of the IUTAM symposium held in Stuttgart, Germany, February 20–23, 2006. Springer Netherlands. https://doi.org/10.1007/978-1-4020-5981-0

[3] Schweizer, B.: IUTAM Symposium on solver-coupling and co-simulation. Proceedings of the IUTAM Symposium on Solver-Coupling and Co-Simulation, Darmstadt, Germany, September 18–20, 2017, IUTAM Bookseries. Springer, Cham. https://doi.org/10.1007/978-3-030-14883-6

[4] Andersson, H., Nordin, P., Borrvall, T., Simonsson, K., Hilding, D., Schill, M., Krus, P., Leidermark, D.: A co-simulation method for system-level simulation of fluid-structure couplings in hydraulic percussion units. Engineering with Computers. **33**(2), 317 (2017). https://doi.org/10.1007/s00366-016-0476-8

[5] MA Fernández, Mullaert, J., Vidrascu, M.: Generalized robin-neumann explicit coupling schemes for incompressible fluid-structure interaction: stability analysis and numerics. International Journal for Numerical Methods in Engineering. **101**(3), 199–229 (2015). https://doi.org/10.1002/nme.4785

[6] Masarati, P., Morandini, M., Quaranta, G., D C handar., Sitaraman, J.: Tightly coupled CFD/multibody analysis of flapping-wing micro-aerial vehicles. 29th AIAA Applied Aerodynamics Conference (2011).

[7] Malhan, R., Baeder, J., Chopra, I., Masarati, P.: CFD-CSD coupled aeroelastic analysis of flexible flapping wings for mav applications: methodology validation. Collection of Technical Papers-AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference (2013). https://doi.org/10.2514/6.2013-1644

[8] Olivier, B., Verlinden, O., Kouroussis, G.: A vehicle/track/soil model using co-simulation between multibody dynamics and finite element analysis. International Journal of Rail Transportation. **8**(2), 135–158 (2020). https://doi.org/10.1080/23248378.2019.1642152

[9] Olivier, B., Verlinden, O., Kouroussis, G.: Effect of applied force cosimulation schemes on recoupled vehicle/track problems. Multibody Syst Dyn. **50**, 337–353 (2020). https://doi.org/10.1007/s11044-020-09748-8

[10] Negrut, D., Melanz, D., Mazhar, H., Lamb, D.: Investigating through simulation the mobility of light tracked vehicles operating on discrete granular terrain. SAE Int. J. Passeng. Cars-Mech. Syst. **6**(1), 369–381 (2013), https://doi.org/10.4271/2013-01-1191

[11] Rahikainen, J., González, F., Naya, M. Á., Sopanen, J., Mikkola, A.: On the cosimulation of multibody systems and hydraulic dynamics. Multibody Syst Dyn. **50**, 143–167 (2020). https://doi.org/10.1007/s11044-020-09727-z

[12] Naya, M., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and co-simulation approaches. Archive of Mechanical Engineering. **58**(2), 223–243 (2011). https://doi.org/10.2478/v10180-011-0016-4

[13] Zhang, R., Zhang, H., Zanoni, A., Wang, Q., Masarati, P.: A tight coupling scheme for smooth/non-smooth multibody co-simulation of a particle damper. Mechanism and Machine Theory. **161**, 104181 (2020). https://doi.org/10.1016/j.mechmachtheory.2020.104181

[14] Spreng, F., Eberhard, P., Fleissner, F.: An approach for the coupled simulation of machining processes using multibody system and smoothed particle hydrodynamics algorithms. Theoretical and Applied Mechanics Letters. **3**, 013005 (2013). https://doi.org/10.1063/2.1301305

[15] Arnold, M.: Multi-rate time integration for large scale multibody system models. In: Eberhard P. (eds) IUTAM Symposium on Multiscale Problems in Multibody System Contacts. IUTAM Bookseries, vol 1, pp. 1–10. Springer, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-5981-0

[16] Arnold, M., Günther, M.: Preconditioned dynamic iteration for coupled differential-algebraic systems. BIT Numerical Mathematics. **41**, 1–25 (2001). https://doi.org/10.1023/A:1021909032551

[17] Schneider, F., Burger, M.: Constraint coupling for flexible multibody systems: stabilization by modified spatial discretization. In: Schweizer B. (eds) IUTAM Symposium on Solver-Coupling and Co-Simulation. IUTAM Bookseries, vol 35, pp. 269–289. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14883-6

[18] Solcia, T., Masarati, P.: Efficient multirate simulation of complex multibody systems based on free software. Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 4: 8th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A and B. Washington, DC, USA. August 28–31, 2011. pp. 29-39. ASME. https://doi.org/10.1115/DETC2011-47306

[19] Papadopoulos, A. V., Leva, A.: A model partitioning method based on dynamic decoupling for the efficient simulation of multibody systems. Multibody Syst Dyn. **34**, 163–190 (2015). https://doi.org/10.1007/s11044-014-9415-x

[20] González, F., Naya, M.Á., Luaces, A., González, M.: On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. Multibody Syst Dyn. **25**, 461–483 (2011). https://doi.org/10.1007/s11044-010-9234-7

[21] Fancello, M., Morandini, M., Masarati, P.: Helicopter rotor sailing by non-smooth dynamics co-simulation. Archive of Mechanical Engineering. **62**(2), 253–268 (2014). http://dx.doi.org/10.2478/meceng-2014-0015

[22] Negrut, D., Serban, R., Mazhar, H., Heyn, T.: Parallel computing in multibody system dynamics: why, when, and how. Journal of Computational and Nonlinear Dynamics. **9**(4), 041007 (2014). https://doi.org/10.1115/1.4027313

[23] Tseng, F. C., Hulbert, G. M.: A gluing algorithm for network-distributed multibody dynamics simulation. Multibody System Dynamics. **6**(4), 377–396 (2001). https://doi.org/10.1023/A:1012279120194

[24] Wang, J., Ma, Z. D., Hulbert, G. M.: A gluing algorithm for distributed simulation of multibody systems. Nonlinear Dynamics. **34**(1-2), 159–188 (2003). https://doi.org/10.1023/B:NODY.0000014558.70434.b0

[25] Guy, B., Asada, H. H.: Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models. Journal of Dynamic Systems Measurement and Control. **126**(1), 1–13 (2004). https://doi.org/10.1115/1.1648307

[26] Schweizer, B., Li, P., Lu, D.: Implicit co-simulation methods: stability and convergence analysis for solver coupling approaches with algebraic constraints. J. Appl. Math. Mech./Z. Angew. Math. Mech. **96**(8), 986–1012 (2016). https://doi.org/10.1002/zamm.201400087

[27] Meyer, T., Li, P.: Implicit co-simulation method for constraint coupling with improved stability behavior. Multibody Syst Dyn. **44**, 135–161 (2018). https://doi.org/10.1007/s11044-018-9632-9

[28] Schweizer, B., Lu, D.: Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. Multibody Syst Dyn. **34**, 129–161 (2015). https://doi.org/10.1007/s11044-014-9422-y

[29] Li, P., Yuan, Q., Lu, D., Meyer, T., Schweizer, B.: Improved explicit co-simulation methods incorporating relaxation techniques. Arch Appl Mech. **90**, 17–46 (2020). https://doi.org/10.1007/s00419-019-01597-y

[30] Schweizer, B., Li, P., Lu, D.: Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches. ASME. J. Comput. Nonlinear Dynam. **10**(5), 051007. https://doi.org/10.1115/1.4028503

[31] Schweizer, B., Li, P., Lu, D., Meyer, T.: Stabilized implicit co-simulation methods: solver coupling based on constitutive laws. Arch Appl Mech. **85**, 1559–1594 (2015). https://doi.org/10.1007/s00419-015-0999-2

[32] Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. Arch Appl Mech. **84**, 1739–1769 (2014). https://doi.org/10.1007/s00419-014-0883-5

[33] Anderson, K. S., Duan, S.: Highly parallelizable low-order dynamics simulation algorithm for multi-rigid-body systems. Journal of Guidance Control and Dynamics. **23**(2), 355–364 (2012). https://doi.org/10.2514/2.4531

[34] Duan, S., Anderson, K.: Parallel Implementation of a Low Order Algorithm for Dynamics of Multibody Systems on a Distributed Memory Computing System. Engineering with Computers. **16**, 96–108 (2000). https://doi.org/10.1007/PL00007191

[35] Sharf, I., D′Eleuterio, G. M. T.: Parallel simulation dynamics for elastic multibody chains. IEEE Transactions on Robotics and Automation. **8**(5), 597–606 (1992). https://doi.org/10.1109/70.163784

[36] Chen, W., Ran, S., Wu, C., Jacobson, B.: Explicit parallel co-simulation approach: analysis and improved coupling method based on H-infinity synthesis. Multibody Syst Dyn **52**, 255–279 (2021). https://doi.org/10.1007/s11044-021-09785-x

[37] Peiret, A., González, F., Kövecses, J., Teichmann, M.: Co-simulation of multibody systems with contact using reduced interface models. ASME. J. Comput. Nonlinear Dynam, **15**(4), 041001 (2020). https://doi.org/10.1115/1.4046052

[38] Antunes, P., Magalhães, H., Ambrósio, J., Pombo, J., Costa, J.: A co-simulation approach to the wheel-rail contact with flexible railway track. Multibody Syst Dyn, **45**, 245–272 (2019). https://doi.org/10.1007/s11044-018-09646-0

[39] Leimkuhler, B.: Relaxation techniques in multibody dynamics. Transactions of the Canadian Society for Mechanical Engineering. **4**, 459–471 (1993). https://doi.org/10.1139/tcsme-1993-0025

[40] Fisette, P., Péterkenne, J. M.: Contribution to parallel and vector computation in multibody dynamics. Parallel Computing. **24**(5-6), 717–728 (1998). https://doi.org/10.1016/S0167-8191(98)00036-2

[41] Lelarasmee, E., Ruehli, A. E., Sangiovanni-Vincentelli, A. L.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. **1**(3), 131–145 (1982). https://doi.org/10.1109/TCAD.1982.1270004

[42] Gear, C. W., Gupta, G. A., Leimkuhler, B.: Automatic integration of Euler-Lagrange equations with constraints. J. Comput. Appl. Math. **12-13**, 77-90,(1985). `https://doi.org/10.1016/0377-0427(85)90008-1`

[43] Ren, H., Yang, K.: A referenced nodal coordinate formulation. Multibody Syst Dyn. **51**, 305–342 (2021). `https://doi.org/10.1007/s11044-020-09750-0`

[44] Orzechowski, G., Matikainen, M. K., Mikkola, A. M.: Inertia forces and shape integrals in the floating frame of reference formulation. Nonlinear Dyn. **88**, 1953–1968 (2017). `https://doi.org/10.1007/s11071-017-3355-y`

[45] Shabana, A. A. *Dynamics of Multibody Systems*, 3rd Edition, Cambridge University Press (2005)

[46] Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Computer Methods in Applied Mechanics and Engineering. **1**(1), 1–16 (1972). `https://doi.org/10.1016/0045-7825(72)90018-7`

[47] Bayo, E., Ledesma, R.: Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. Nonlinear Dyn. **9**, 113–130 (1996). `https://doi.org/10.1007/BF01833296`

[48] Bauchau, O. A.: A self-stabilized algorithm for enforcing constraints in multibody systems. International Journal of Solids and Structures, **40**(13–14), 3253–3271 (2003). `https://doi.org/10.1016/S0020-7683(03)00159-8`

[49] Gear, C. W.: Towards explicit methods for differential algebraic equations. BIT. **46**, 505–514 (2005) `https://doi.org/10.1007/s10543-006-0068-x`

[50] Wehage, R. A., Haug, E. J.: Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, Trans. ASME, J. Mech. Design. **134**, 247-255 (1982). `https://doi.org/10.1115/1.3256318`

[51] Bauchau, O. A., Laulusa, A.: Review of contemporary approaches for constraint enforcement in multibody systems. Journal of Computational and Nonlinear Dynamics. **3**(1), 011005 (2008). `https://doi.org/10.1115/1.2803257`

[52] Braun, D. J., Goldfarb, M.: Eliminating constraint drift in the numerical simulation of constrained dynamical systems. Comput. Methods Appl. Mech. Engrg. **198**(37-40), 3151–3160 (2009). `https://doi.org/10.1016/j.cma.2009.05.013`

[53] Abdulle, A.: Fourth order chebyshev methods with recurrence relation. SIAM J. Sci. Comput. **23**(6), 2041–2054 (2002). `https://doi.org/10.1137/S1064827500379549`

[54] Sonneville. V., Brüls, O.: A formulation on the special Euclidean group for dynamic analysis of multibody systems. ASME J. Comput. Nonlinear Dynam. Oct 2014, **9**(4), 041002. `https://doi.org/10.1115/1.4026569`

[55] Rosenbrock, H. H.: Some general implicit processes for the numerical solution of differential equations. The Computer Journal, **5**(4), 329–330 (1963). https://doi.org/10.1093/comjnl/5.4.329

[56] Masarati, P., Morandini, M., Mantegazza, P.: An efficient formulation for general-purpose multibody/multiphysics analysis. ASME J. Comput. Nonlinear Dyn. **9(4)**, 041001 (2014). https://doi.org/10.1115/1.4025628

[57] Ghiringhelli, G. L., Masarati, P., Mantegazza, P.: A multibody implementation of finite volume C^0 beams. AIAA Journal, **38**(1), 131-138 (2000). https://doi.org/10.2514/2.933

[58] Bauchau, O. A., Betsch, P., Cardona, A., Gerstmayr, J., Jonker, B., Masarati, P., Sonneville, V.: Validation of flexible multibody dynamics beam formulations using benchmark problems. Multibody System Dynamics, **37**(1), 29-48 (2016). https://doi.org/10.1007/s11044-016-9514-y

[59] Arnold, M., Brüls, O.: Convergence of the generalized-α scheme for constrained mechanical systems. Multibody System Dynamics, **18**, 185–202 (2007). https://doi.org/10.1007/s11044-007-9084-0

[60] Brüls, O., Cardona. A., Arnold. M.: Lie group generalized-α time integration for constrained flexible multibody systems. Mechanism and Machine Theory, **48**, 121–137 (2012). https://doi.org/10.1016/j.mechmachtheory.2011.07.017

[61] Ren, H.: A simple absolute nodal coordinate formulation for thin beams with large deformations and large rotations. Journal of Computational and Nonlinear Dynamics, **10(6)**, 061005 (2015). https://doi.org/10.1016/j.mechmachtheory.2011.07.017