



POLITECNICO
MILANO 1863

[RE.PUBLIC@POLIMI](#)

Research Publications at Politecnico di Milano

This is the published version of:

R. Zhang, H. Zhang, A. Zanoni, A. Tasora, P. Masarati
Explicit Smooth/nonsmooth Cosimulation Using Kinematic Constraints
Multibody System Dynamics, Published online 06/06/2022
doi:10.1007/s11044-022-09829-w

The final version is available online at: <https://doi.org/10.1007/s11044-022-09829-w>

The project leading to this application has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 687023. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the

Union




Permanent link to this version

<http://hdl.handle.net/11311/1217520>



Explicit smooth/nonsmooth cosimulation using kinematic constraints

Runsen Zhang^{1,2,3}  · Huimin Zhang⁴ · Andrea Zanoni¹ · Alessandro Tasora⁵ · Pierangelo Masarati¹

Received: 10 April 2021 / Accepted: 9 May 2022
© The Author(s) 2022

Abstract

An explicit cosimulation scheme is developed to study the coupling of smooth and nonsmooth systems using kinematic constraints. Using the force-displacement decomposition, the coupling constraints are formulated at the velocity level, to preserve consistency with the impulse-momentum equations for frictional contacts in the nonsmooth solver, which however potentially leads to instability of the explicit cosimulation. To improve the stability of the cosimulation without affecting the format of the coupling constraints, guidelines for the modification of the prescribed motion are developed following the spirit of Baumgarte's stabilization technique and the characteristics of the proposed integration scheme, which prescribes a combination of position, velocity, and acceleration to the constrained bodies. Using modified inputs, the stability of the cosimulation is tested using a rigidly connected two-mass oscillator model, which shows clear improvement compared to that with unaltered inputs. The performances of the cosimulation with modified inputs are further illustrated using a double-pendulum system and a complex flexible multibody system coupled with a particle damper. It follows that cosimulation results well agree with those obtained using monolithic simulation or simplified models, verifying the explicit smooth/nonsmooth cosimulation. The results also show a higher efficiency of the explicit cosimulation scheme, which requires much less computational time to obtain similar results, compared to the implicit smooth/nonsmooth cosimulation.

Keywords Explicit cosimulation · Smooth/nonsmooth coupled system · Stabilization technique · Cosimulation stability

1 Introduction

Multibody dynamics is often used to simulate multiphysics and multidomain problems. Although a monolithic approach is sometimes feasible (for example, coupled helicopter aeromechanics and pilot biomechanics in [1, 2]), cosimulation techniques are widely used to coordinate different solvers [3–7] and to parallelize the simulation process [8–10]. In the present work, we address the application of explicit cosimulation techniques to smooth/nonsmooth coupled multibody systems.

The cosimulation of nonsmooth systems has been studied by several authors, most of them focusing on smooth/nonsmooth subsystems connected by contact forces calculated

Extended author information available on the last page of the article

by penalty methods or contact laws and then sent to the subsystems [11]. This approach is commonly used in the cosimulation of discrete element methods (DEM) or nonsmooth methods [12, 13] and multibody dynamics (MBD) [4, 9, 14, 15]. Apart from connecting subsystems through contact forces, in some studies, smooth and nonsmooth subsystems are linked by ideal kinematic constraints, for example, mechanical systems mounting particle dampers [16] and nonsmooth systems coupled with other physical modules [17]. In these cases, although contact forces can still be employed as coupling variables, an alternative choice is to decompose the overall system at the kinematic constraints and to send reaction forces/moments to other subsystems. The use of reaction forces/moments avoids calculating the sum of contact forces, which may be time-consuming if a large number of nonsmooth events takes place on the coupling body. This approach can also help confine the details of the solution process in nonsmooth subsystems.

This work adopts the latter approach, namely smooth/nonsmooth cosimulation with kinematic constraints. We first discuss the coupled smooth systems with kinematic constraints. In this case the whole system can be formulated using Newton–Euler’s equations as a set of differential-algebraic equations (DAEs) [18]. In cosimulation a common way is to decompose the whole DAEs in several sets of ordinary differential equations (ODEs) governing the motion of each subsystem and a set of algebraic equations representing the kinematic constraints between subsystems. Using the decomposition, Kubler et al. [18, 19] provided two implicit schemes, also called “iterative schemes,” which employ an approximate Jacobian matrix to guarantee the local convergence of cosimulation iterations. In [20] a semi-implicit cosimulation scheme including prediction and correction procedures was proposed, which shows advantages in terms of stability compared to explicit cosimulation schemes. The prediction–correction procedure incorporates different stabilization methods, such as the projection technique [21], Baumgarte’s stabilization method [22], the weighted multiplier approach [22], and the Gear–Gupta–Leimkuhler (GGL) method [23], to enforce the constraints and to yield reaction forces/moments. The resulting forces/moments are then sent to the subsystems to continue the co-simulation. Apart from integrating stabilization methods in the semi-implicit scheme, to achieve higher convergence rates, a technique that updates higher-order approximation polynomials in each macrostep [24] was also established to enforce the coupling conditions at all levels: position, velocity, and acceleration.

Although the above-mentioned semiimplicit/implicit cosimulation schemes show satisfactory stability, explicit solver coupling is still widely used, owing to its advantage in terms of efficiency. Gu et al. [25, 26] developed an explicit cosimulation scheme, which adopts Baumgarte’s stabilization method to deal with the coupling algebraic equations, where index- r DAEs are reformulated as index-1 DAEs to generate reaction forces/moments for the subsystems. In [27] a force-displacement cosimulation with kinematic constraints was also explicitly implemented to avoid constraint drift problems. This study pointed out that in the cosimulation platform, using the index-1 monolithic equations to calculate constraint forces may result in an unacceptable amount of computation when the coupled system is large, whereas formulating the constraint forces as functions of acceleration vectors generates algebraic loops in the cosimulation, even spoiling the zero-stability of cosimulation schemes in some cases [27]. Apart from calculating reaction forces in the cosimulation platform, if subsystems can produce reaction forces, then the coupling constraints can also be modeled in subsystems by enforcing kinematics. Using a nonsmooth method that can deal with bilateral constraints [28, 29], an embedded force-displacement cosimulation scheme was developed, where the coupling constraints can be formulated in one subsystem to generate reaction forces/moments as outputs [7]. However, using this approach, an implicit cosimulation scheme with an iterative solution process is required to enhance convergence and enforce algorithmic stability [7].

For coupled smooth/nonsmooth systems with kinematic constraints, if an explicit force-displacement cosimulation scheme [7] is employed in addition to the risk of losing stability, then another problem that may occur is the inconsistency of kinematic constraints between the monolithic DAEs and the nonsmooth methods. In fact, in nonsmooth time-stepping methods [17, 28, 30–32], kinematic constraint equations are usually formulated at the velocity level for consistency with the nonsmooth dynamic equations at the impulse-momentum level, whereas in monolithic DAEs and cosimulation, constraint equations are expected to be satisfied at position, velocity, and acceleration levels at the synchronization time [21, 22]. This inconsistency cannot be eliminated when subsolvers are expected to remain opaque, as is a common practice when cosimulating between existing well-established solvers, and although just alternatively coupling positions or velocities would be an option, the stability of the cosimulation scheme would be poor in general cases.

In this paper, we develop an explicit cosimulation scheme for smooth/nonsmooth coupled multibody systems with kinematic constraints. A force-displacement decomposition technique is employed to divide the whole system at coupling kinematic constraints into smooth subsystems with external dynamic variables, i.e., prescribed forces, and nonsmooth subsystems with external kinematic variables, i.e., prescribed kinematics. In smooth subsystems, dynamic variables are applied to bodies at markers of coupling constraints, whereas in nonsmooth subsystems, kinematic variables are enforced to bodies through rheonomic constraints at the velocity level. To guarantee the imposition of all coupling kinematics, namely position, velocity, and acceleration, to the bodies in the nonsmooth subsystems and at the same time to maintain the confinement of the nonsmooth solver, the input kinematics resulting from the smooth subsystems are modified using Baumgarte’s stabilization method [33] and the properties of the integrators in the nonsmooth subsystems. As in other studies [20], a two-mass oscillator is used to assess how the modified inputs improve algorithmic stability. The analysis shows that the modified inputs can enlarge the stability region, compared to directly using the original constraint equations at the velocity level in the nonsmooth subsystems. Finally, the capabilities of the explicit cosimulation scheme are further assessed by applying it to a double pendulum and to a smooth/nonsmooth coupled system consisting of a flexible crank-slider mechanism actuating a beam that carries a particle damper.

Specifically, in Sect. 2, we decompose a coupled system into smooth and nonsmooth subsystems. For each subsystem, we give formulations and modified input–output schemes. In Sect. 3, we introduce the cosimulation strategy, where an explicit cosimulation scheme and its implementation are discussed. In Sect. 4, a linear two-mass oscillator is cosimulated to compare the effect of different input formats on algorithmic stability, and two other numerical examples are also analyzed to show the effectiveness of the smooth/nonsmooth cosimulation with modified inputs. In Sect. 5, we finally draw conclusions.

2 Formulation of coupled systems

2.1 System decomposition

As shown in Fig. 1(a), a coupled smooth/nonsmooth system consists of a set of bodies S_i ($i = 0, 1, 2, \dots$) whose dynamics are formulated as smooth, i.e., they are assumed to be unaffected by impacts or friction (called for simplicity “smooth bodies”), and another set of bodies NS_j ($j = 0, 1, 2, \dots$) that can be subject to nonsmooth events (called “nonsmooth bodies”). According to the force-displacement decomposition technique [21, 27], this coupled system can be divided into smooth and nonsmooth subsystems at bilateral constraints

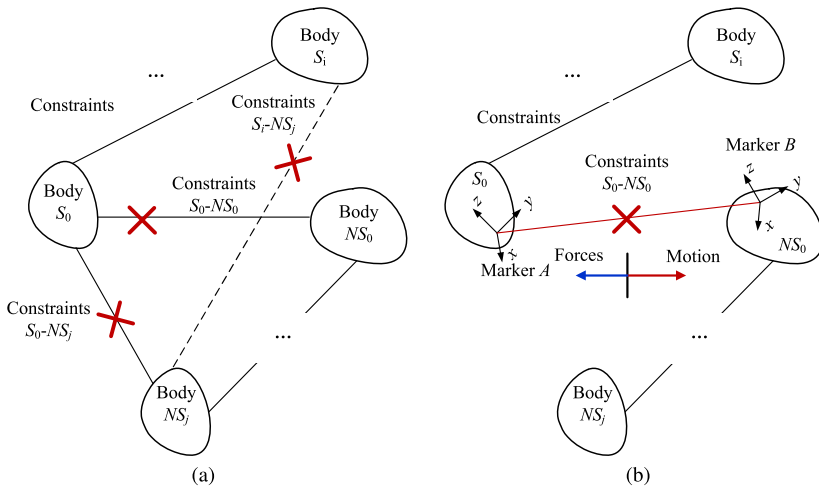


Fig. 1 Decomposition of coupled system: (a) with multiple connections, (b) with one coupling connection at constraints $S_0 - NS_0$ (Marker A-Marker B)

that connect sets of smooth and nonsmooth bodies. For clarity of explanation, let us discuss the coupled system of Fig. 1(b). A smooth subsystem and a nonsmooth subsystem are connected by constraints between markers A and B, which are rigidly attached to bodies S_0 and NS_0 , respectively. By the force-displacement decomposition this coupled system is divided into a smooth subsystem and a nonsmooth subsystem at the coupling constraints. The smooth subsystem generates the motion of marker A as output, whereas the nonsmooth subsystem supplies to the smooth subsystem the reaction forces and moments that act on marker A.

2.2 Formulation of subsystems

2.2.1 Formulation of smooth subsystems

The smooth subsystem consisting of bodies S_i is modeled as a generic multibody system, which can be formulated as a set of DAEs using the physical coordinates \mathbf{q}_1 as in [34, 35]:

$$\mathbf{M}_1 \dot{\mathbf{v}}_1 + \mathbf{g}_{1/q_1}^T \boldsymbol{\lambda}_1 + \mathbf{g}_{co/q_1}^T \boldsymbol{\lambda}^{co} - \mathbf{f}_1(\mathbf{q}_1, \mathbf{v}_1, t) = \mathbf{0}, \tag{1a}$$

$$\mathbf{g}_1(\mathbf{q}_1, t) = \mathbf{0}, \tag{1b}$$

where \mathbf{M}_1 is the mass matrix, $\mathbf{v}_1 = \dot{\mathbf{q}}_1$ represents the generalized velocity coordinate vector, $t \geq 0$ denotes the time, vectors \mathbf{g}_1 and \mathbf{g}_{co} indicate the constraints in the smooth subsystem and the coupling constraint $S_0 - NS_0$ between the subsystems, as shown in Fig. 1(b), the latter being defined later in Eqs. (2a)–(2e) as part of the nonsmooth subsystem dynamics, vectors $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}^{co}$ represent the Lagrange multipliers of the corresponding constraints, vector \mathbf{f}_1 contains the generalized external forces, the over dot denotes the derivative with respect to time, and the operator $(\cdot)_{/x}$ denotes the partial derivative with respect to \mathbf{x} .

2.2.2 Formulation of nonsmooth subsystem

Nonsmooth methods [13, 30, 32, 36] are effective tools to deal with nonsmooth dynamics. In nonsmooth methods, bodies are assumed to be rigid, and nonsmooth events among bodies are described as frictional unilateral constraints, such as the Coulomb friction and impacts, which are transformed to nonlinear/linear complementarity problems (N/LCPs) or projection functions to be solved [37]. The nonsmooth methods are extended to systems with bilateral constraints [29, 38, 39], where bilateral constraints are formulated at the velocity level with different stabilization methods, to enforce the bilateral constraints at both position and velocity levels. In the present work, we use the nonsmooth method proposed by Tasora and Anitescu [29], based on cone complementarity problems (CCPs). The method uses a semiimplicit Euler integrator; bilateral constraints are modified as a combination of constraints at position and velocity levels. The nonsmooth method is included in the open-source C++ library Project Chrono.¹

Using the CCP-based method [28, 29], we formulate a nonsmooth subsystem with coupling constraints as

$$\mathbf{M}_2 \dot{\mathbf{v}}_2 + \mathbf{g}_{2/q_2}^T \boldsymbol{\lambda}_2 + \mathbf{g}_{co/q_2}^T \boldsymbol{\lambda}^{co} + \sum_{\pi_i \in \Pi} \mathbf{D}_{\pi_i} \mathbf{p}_{\pi_i} = \mathbf{f}_2(\mathbf{q}_2, \mathbf{v}_2, t), \tag{2a}$$

$$\mathbf{g}_2(\mathbf{q}_2, t) = \mathbf{0}, \tag{2b}$$

$$\mathbf{g}_{co}(\mathbf{q}_1, \mathbf{q}_2) = \mathbf{0}, \tag{2c}$$

$$\pi_i \in \Pi : 0 \leq p_{\mathbf{n}_{\pi_i}} \perp \Phi_{\mathbf{n}_{\pi_i}} \geq 0, \tag{2d}$$

$$\left(p_{\mathbf{u}_{\pi_i}}, p_{\mathbf{w}_{\pi_i}} \right) = \underset{\|p_{\mathbf{u}_{\pi_i}} \mathbf{u}_{\pi_i} + p_{\mathbf{w}_{\pi_i}} \mathbf{w}_{\pi_i}\| \leq \mu_{\pi_i} p_{\mathbf{n}_{\pi_i}}}{\arg \min} \mathbf{v}_{T_{\pi_i}}^T \left(p_{\mathbf{u}_{\pi_i}} \mathbf{u}_{\pi_i} + p_{\mathbf{w}_{\pi_i}} \mathbf{w}_{\pi_i} \right), \tag{2e}$$

where \mathbf{M}_2 is the mass matrix, the vectors \mathbf{q}_2 and \mathbf{v}_2 are generalized position and velocity coordinates, respectively, the vector \mathbf{g}_2 denotes the bilateral constraints in the nonsmooth subsystem, and \mathbf{g}_{co} is the constraint $S_0 - NS_0$ between the subsystems, representing the interfaces used to prescribe the motion to the nonsmooth bodies, \mathbf{g}_{2/q_2} and \mathbf{g}_{co/q_2} are the corresponding Jacobian matrices, the vectors $\boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}^{co}$ represent the Lagrange multipliers of the corresponding constraints, the vector \mathbf{f}_2 contains the generalized external forces, the set Π collects all closed contacts, the index π_i refers to a generic element of the set Π , for the contact π_i , the local reference is defined by three orthogonal vectors \mathbf{n}_{π_i} , \mathbf{u}_{π_i} , and \mathbf{w}_{π_i} at the contact point, where \mathbf{n}_{π_i} denotes the normal direction, whereas \mathbf{u}_{π_i} and \mathbf{w}_{π_i} are orthogonal vectors representing the tangential directions, \mathbf{p}_{π_i} is the vector of the generalized forces generated by the frictional contacts, $p_{\mathbf{n}_{\pi_i}}$, $p_{\mathbf{u}_{\pi_i}}$, and $p_{\mathbf{w}_{\pi_i}}$ are the three components of \mathbf{p}_{π_i} along directions \mathbf{n}_{π_i} , \mathbf{u}_{π_i} , and \mathbf{w}_{π_i} , respectively, $v_{\mathbf{n}_{\pi_i}}$ and $\mathbf{v}_{T_{\pi_i}}^T = [v_{\mathbf{u}_{\pi_i}}, v_{\mathbf{w}_{\pi_i}}]$ are the normal component and the tangential projection of the velocity vector at the contact point, and $\Phi_{\mathbf{n}_{\pi_i}}$ denotes the normal distance at the contact π_i .

To improve notation readability, we further assume that $\mathbf{v}_2 = \dot{\mathbf{q}}_2$, although in the nonsmooth method [28, 29] the rotation is described by quaternions in \mathbf{q}_2 , which are updated using an exponential map to preserve their unimodularity, whereas the angular velocity in the local reference is used in \mathbf{v}_2 .

¹<http://projectchrono.org/>, last accessed by March 2021.

2.2.3 Discrete equations

Since both smooth and nonsmooth solvers are assumed to model the respective portions of the problem as sets of rigid bodies, before discretizing the equations, we rewrite the coupling constraints as functions of the generalized coordinates related to the interface bodies S_0 and NS_0 . As shown in Fig. 1(b), the constraint $S_0 - NS_0$, written as \mathbf{g}_{co} in Eq. (2c), only involves the motion of body S_0 in the smooth subsystem and of body NS_0 in the nonsmooth subsystem. Therefore the constraint \mathbf{g}_{co} in Eq. (2c) can be rewritten as

$$\mathbf{g}_{co}(\mathbf{q}_1, \mathbf{q}_2) = \mathbf{g}_{co}(\mathbf{q}_1^{co}, \mathbf{q}_2^{co}) = \mathbf{0}, \tag{3}$$

where \mathbf{q}_1^{co} and \mathbf{q}_2^{co} denote the physical coordinates of the coupling body, S_0 and NS_0 , in the respective subsystems. According to Eq. (3), Eq. (1a) can be rewritten as

$$\mathbf{M}_1 \dot{\mathbf{v}}_1 + \mathbf{g}_{1/q_1}^T \boldsymbol{\lambda}_1 + \begin{bmatrix} \mathbf{g}_{co/q_1^{co}}^T \\ \mathbf{0} \end{bmatrix} \boldsymbol{\lambda}^{co} - \mathbf{f}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1, t) = \mathbf{0}. \tag{4}$$

In Eqs. (1a), (1b) and (4), DAEs are solved using the formulation proposed in [34], i.e., discretized at time t_{n+1} as

$$\mathbf{M}_1 \ddot{\mathbf{q}}_{1,n+1} + \mathbf{g}_{1/q_1}^T \boldsymbol{\lambda}_{1,n+1} + \begin{bmatrix} \mathbf{g}_{co/q_1^{co}}^T \\ \mathbf{0} \end{bmatrix} \boldsymbol{\lambda}_{n+1}^{co} - \mathbf{f}_{1,n+1}(\mathbf{q}_{1,n+1}, \dot{\mathbf{q}}_{1,n+1}, t_{n+1}) = \mathbf{0}, \tag{5a}$$

$$\mathbf{g}_1(\mathbf{q}_{1,n+1}, t_{n+1}) = \mathbf{0}. \tag{5b}$$

The state variables $\mathbf{x} = [\mathbf{q}_1^T, \dot{\mathbf{q}}_1^T, \boldsymbol{\lambda}_1^T]^T$ are predicted and updated at each step according to an implicit linear two-step scheme [34, 39] as

$$\mathbf{x}_{n+1} = \sum_{i=1,2} a_i \mathbf{x}_{n+1-i} + \Delta t \sum_{i=0,1,2} b_i \dot{\mathbf{x}}_{n+1-i}, \tag{6}$$

which has second-order accuracy and is unconditionally stable with tunable algorithmic dissipation. The coefficients are defined as functions of the asymptotic spectral radius ρ_∞ ($\rho_\infty \in [0, 1]$) and are expressed as [34]

$$a_1 = 1 - \frac{3(1 - \rho_\infty)^2 + 4(2\rho_\infty - 1)}{4 - (1 - \rho_\infty)^2}, \tag{7a}$$

$$a_2 = \frac{3(1 - \rho_\infty)^2 + 4(2\rho_\infty - 1)}{4 - (1 - \rho_\infty)^2} = 1 - a_1, \tag{7b}$$

$$b_0 = \frac{(1 - \rho_\infty)^2}{2[4 - (1 - \rho_\infty)^2]} + \frac{1}{2}, \tag{7c}$$

$$b_1 = \frac{3(1 - \rho_\infty)^2 + 4(2\rho_\infty - 1)}{2[4 - (1 - \rho_\infty)^2]} + \frac{1}{2} - \frac{(1 - \rho_\infty)^2}{4 - (1 - \rho_\infty)^2}, \tag{7d}$$

$$b_2 = \frac{3(1 - \rho_\infty)^2 + 4(2\rho_\infty - 1)}{2[4 - (1 - \rho_\infty)^2]} + \frac{(1 - \rho_\infty)^2}{2[4 - (1 - \rho_\infty)^2]}. \tag{7e}$$

This implicit linear two-step scheme for generic multibody systems is implemented in the free, general-purpose multibody software MBDyn,² which in the present work is used to model the smooth portion of the system. Although the complete orientation matrix of the interface bodies is exchanged with the subsystems, in the solution process the incremental Cayley–Gibbs–Rodriguez (CGR) rotation parameters are used as state variables to update the orientation of the bodies [34]. For ease of notation, we use a generalized coordinate set \mathbf{q}_1 to represent displacements and rotations in the smooth subsystem. In the integration procedure the dynamics equations (5a) and (5b) can be rewritten as [34]

$$\mathbf{Res}(\mathbf{x}_{n+1}, \dot{\mathbf{x}}_{n+1}, t_{n+1}) = 0, \tag{8}$$

where the coupling Lagrange multiplier λ^{co} does not explicitly appear because it is determined by the inputs, which are only related to the time for the noniterative cosimulation or to the iterations for the iterative cosimulation. To solve the dynamics problem (8), we employ the prediction–correction approach [34], by which $\dot{\mathbf{x}}_{n+1}$ is predicted first, written as $\dot{\mathbf{x}}_{n+1}^{(0)}$, and then is substituted into the implicit two-step scheme to calculate the predicted \mathbf{x}_{n+1} , written as $\mathbf{x}_{n+1}^{(0)}$. Then, with the perturbation of Eq. (6) expressed as

$$\partial \mathbf{x}_{n+1} = \Delta t b_0 \partial \dot{\mathbf{x}}_{n+1}, \tag{9}$$

the correction procedure is given as [34]

$$(\Delta t b_0 \mathbf{Res}_{/\mathbf{x}} + \mathbf{Res}_{/\dot{\mathbf{x}}}) \partial \dot{\mathbf{x}} = -\mathbf{Res}(\mathbf{x}_{n+1}^{(j)}, \dot{\mathbf{x}}_{n+1}^{(j)}, t_{n+1}^{(j)}), \tag{10a}$$

$$\dot{\mathbf{x}}_{n+1}^{(j+1)} = \dot{\mathbf{x}}_{n+1}^{(j)} + \partial \dot{\mathbf{x}}, \tag{10b}$$

$$\mathbf{x}_{n+1}^{(j+1)} = \mathbf{x}_{n+1}^{(j)} + \Delta t b_0 \partial \dot{\mathbf{x}}. \tag{10c}$$

The dynamics equations of the nonsmooth subsystem (2a)–(2e) are solved at the impulse-momentum level [29], since under the assumption that all bodies are rigid, frictional contacts may lead to inconsistent or indeterminate solutions at the acceleration-force level [29, 40]. At the impulse-momentum level the dynamics equations are discretized by a half-implicit Euler scheme at time t_{n+1} [28, 29] as

$$\mathbf{q}_{2,n+1} = \mathbf{q}_{2,n} + \Delta t \dot{\mathbf{q}}_{2,n+1}, \tag{11a}$$

$$\mathbf{M}_2 (\mathbf{v}_{2,n+1} - \mathbf{v}_{2,n}) + \mathbf{g}_{2/\mathbf{q}_{2,n}}^T \tilde{\lambda}_{2,n+1} + \mathbf{g}_{\text{co}/\mathbf{q}_{2,n}}^T \tilde{\lambda}_{n+1}^{\text{co}} + \sum_{\pi_i \in \Pi} \mathbf{D}_{\pi_i,n} \tilde{\mathbf{p}}_{\pi_i,n+1} = \tilde{\mathbf{f}}_{2,n}, \tag{11b}$$

$$\frac{\mathbf{g}_{2,n}}{\Delta t} + \mathbf{g}_{2/\mathbf{q}_{2,n}} \dot{\mathbf{q}}_{2,n+1} + \mathbf{g}_{2/t,n} = \mathbf{0}, \tag{11c}$$

$$\frac{\mathbf{g}_n^{\text{co}}}{\Delta t} + \mathbf{g}_{\text{co}/\mathbf{q}_{2,n}^{\text{co}}} \dot{\mathbf{q}}_{2,n+1}^{\text{co}} + \mathbf{g}_{\text{co}/\mathbf{q}_{1,n}^{\text{co}}} \dot{\mathbf{q}}_{1,n+1}^{\text{co}} = \mathbf{0}, \tag{11d}$$

where $\tilde{\lambda}_{2,n+1} \equiv \Delta t \lambda_{2,n+1}$, $\tilde{\lambda}_{n+1}^{\text{co}} \equiv \Delta t \lambda_{n+1}^{\text{co}}$, $\tilde{\mathbf{p}}_{\pi_i,n+1} \equiv \Delta t \mathbf{p}_{\pi_i,n+1}$, $\tilde{\mathbf{f}}_{2,n} \equiv \Delta t \mathbf{f}_{2,n}$, and $\mathbf{g}_{2/t} \equiv \partial \mathbf{g}_2 / \partial t$ are defined [28, 29]. Besides, there are two aspects related to the above discrete equations:

²<https://www.mbdyn.org>, last accessed by March 2021.

- concerning Baumgarte’s stabilization method [33], the discretized bilateral constraints \mathbf{g}_{co} and \mathbf{g}_2 are prescribed as a combination of their expression at the position and velocity levels, where the stabilization term $\mathbf{g}_{2,n}/\Delta t$ is employed to avoid drifting away from the constraint manifold while, at the same time, reducing the index of the corresponding set of DAEs [28, 29]. Since the integration scheme is semiexplicit, an explicit term $\mathbf{g}_{2/1,n}$ is used in the equation [28];
- the discrete equations involving unilateral constraints are not listed for simplicity since we focus on the interface with the other subsystem, namely the coupling constraint \mathbf{g}_{co} , in the cosimulation, where unilateral constraints are assumed to be not involved.

In this nonsmooth solution approach [29], the velocity vector $\dot{\mathbf{q}}_{2,n+1}$ is solved using the discrete formulation of Eq. (11a)–(11d) using CCPs [41]; the position and acceleration vectors are updated as [29]

$$\mathbf{q}_{2,n+1} = \mathbf{q}_{2,n} + \Delta t \cdot \dot{\mathbf{q}}_{2,n+1}, \tag{12a}$$

$$\ddot{\mathbf{q}}_{2,n+1} = \frac{\dot{\mathbf{q}}_{2,n+1} - \dot{\mathbf{q}}_{2,n}}{\Delta t}. \tag{12b}$$

2.3 Input–output scheme of decomposed subsystems

In the cosimulation, λ_{n+1}^{co} in Eq. (5a) for the smooth subsystem and $\mathbf{q}_{1,n+1}^{co}$ and its derivatives in Eq. (11d) for the nonsmooth subsystem are obtained from the other subsystem. As a consequence, these two equations are modified as

$$\mathbf{M}_1 \dot{\mathbf{v}}_1 + \mathbf{g}_{1/q_1}^T \lambda_1 + \begin{bmatrix} -\mathbf{g}_{co/q_1^{co}}^T \\ \mathbf{0} \end{bmatrix} \mathbf{u}_1 - \mathbf{f}_1(\mathbf{q}_1, \mathbf{v}_1, t) = \mathbf{0} \tag{13}$$

and

$$\frac{\mathbf{g}_n^{co}(\mathbf{q}_2^{co}, \mathbf{u}_{21})}{\Delta t} + \mathbf{g}_{co/q_2^{co},n} \dot{\mathbf{q}}_{2,n+1}^{co} - \mathbf{u}_{22} = \mathbf{0}, \tag{14}$$

where the vectors \mathbf{u}_1 and $\mathbf{u}_2 = [\mathbf{u}_{21}^T, \mathbf{u}_{22}^T]^T$ are the inputs of the smooth and nonsmooth subsystems, respectively; \mathbf{u}_1 and \mathbf{u}_2 are related to the Lagrange multipliers λ^{co} and the coupling state variables $\mathbf{q}_1^{co}, \dot{\mathbf{q}}_1^{co}, \ddot{\mathbf{q}}_1^{co}$, respectively, expressed as

$$\begin{cases} \mathbf{u}_1 = \mathbf{y}_2 = Y_2(\lambda^{co}), \\ \mathbf{u}_2 = \mathbf{y}_1 = Y_1(\mathbf{q}_1^{co}, \dot{\mathbf{q}}_1^{co}, \ddot{\mathbf{q}}_1^{co}), \end{cases} \tag{15}$$

where \mathbf{y}_i ($i = 1, 2$) are the outputs of the i th subsystem, and Y_i are functions of the corresponding subsystem state variables.

To illustrate how inputs are constructed, in this section, we employ a simple coupling joint $\mathbf{g}_{co} \in \mathbb{R}^6$, which clamps the generalized coordinates of bodies S_0 and NS_0 together, expressed as

$$\mathbf{g}_{co}(\mathbf{q}_1^{co}, \mathbf{q}_2^{co}) = \mathbf{q}_2^{co} - \mathbf{q}_1^{co} = \mathbf{0}. \tag{16}$$

It is worth mentioning that the interface constraint between bodies S_0 and NS_0 is not actually enforced by prescribing Eq. (16). That expression is merely intended to show the relationship between reaction forces/moments and kinematic variables in the smooth subsystem.

Reaction forces and moments in the global reference are sent to the smooth subsystem, so that the input and the Jacobian matrix of the smooth subsystem can be formulated as

$$\mathbf{u}_1 = -\lambda^{\text{co}}, \tag{17a}$$

$$\mathbf{g}_{\text{co}}/\mathbf{q}_1^{\text{co}} = -\mathbf{E}_6, \tag{17b}$$

where $\mathbf{E}_6 \in \mathbb{R}^{6 \times 6}$ is the identity matrix.

In the solution process, the constraint \mathbf{g}_{co} is formulated in the nonsmooth subsystem with respect to a moving local reference defined by the inputs \mathbf{u}_{21} .

2.3.1 Original inputs

In the nonsmooth subsystem the interfaces for the input of the motion are designed according to the discrete bilateral constraints in Eq. (14) as (with reference to the class `ChLinkMotionImposed` in Project Chrono)

$$\frac{\mathbf{q}_{2,n}^{\text{co}} - \mathbf{u}_{21}}{\Delta t} + \dot{\mathbf{q}}_{2,n+1}^{\text{co}} - \mathbf{u}_{22} = \mathbf{0}. \tag{18}$$

By comparing the interfaces with the coupling constraints $\mathbf{g}_{\text{co}} = \mathbf{0} \in \mathbb{R}^6$ of Eq. (16), which are discretized as

$$\frac{\mathbf{q}_{2,n}^{\text{co}} - \mathbf{q}_{1,n}^{\text{co}}}{\Delta t} + \dot{\mathbf{q}}_{2,n+1}^{\text{co}} - \dot{\mathbf{q}}_{1,n+1}^{\text{co}} = \mathbf{0}, \tag{19}$$

we obtain the corresponding inputs expressed as

$$\begin{cases} \mathbf{u}_{21} = \mathbf{q}_{1,n}^{\text{co}}, \\ \mathbf{u}_{22} = \dot{\mathbf{q}}_{1,n+1}^{\text{co}}. \end{cases} \tag{20}$$

It is worth noticing that whereas the discrete velocity constraint is evaluated at time t_{n+1} , the discrete position constraint is evaluated at time t_n .

Using this set of inputs, we solve the coupling constraint \mathbf{g}_{co} in the nonsmooth subsystem, which linearly combines the constraints at position and velocity levels, although not enforced at the same time step. However, in cosimulation, this combination of the constraint formulation may jeopardize algorithmic stability. A more stable scheme is needed to guarantee adequate enforcement of \mathbf{g}_{co} at the position, velocity, and acceleration levels.

We further call this set of inputs “original inputs.”

2.3.2 Inputs according to Baumgarte’s method

To improve the algorithmic stability of the cosimulation and, at the same time, avoid exposing the details of the nonsmooth solver solution process, we redesign the inputs of Eq. (20). We define the new inputs with the specific objective of satisfying the constraints obtained from Baumgarte’s stabilization method [33] formulated as

$$\alpha (\mathbf{q}_{2,n+1}^{\text{co}} - \mathbf{q}_{1,n+1}^{\text{co}}) + \beta (\dot{\mathbf{q}}_{2,n+1}^{\text{co}} - \dot{\mathbf{q}}_{1,n+1}^{\text{co}}) + \gamma (\ddot{\mathbf{q}}_{2,n+1}^{\text{co}} - \ddot{\mathbf{q}}_{1,n+1}^{\text{co}}) = \mathbf{0}, \tag{21}$$

where $\alpha > 0$, $\beta > 0$, and $\gamma > 0$ are the parameters of Baumgarte’s stabilization method. In general, the choice

$$\alpha = \frac{2}{\Delta t^2}, \quad \beta = \frac{2}{\Delta t}, \quad \gamma = 1 \tag{22}$$

provides an adequate (i.e., accurate and fast) enforcement of the constraint at the position level [42]. It is worth noticing that in Eq. (21), $\gamma = 0$ generates a formulation without accelerations, namely index-2 DAEs for the coupled system, and $\beta = \gamma = 0$ yields index-3 DAEs.

In the nonsmooth subsystem, can be rewrite the interface of Eq. (18) as

$$\dot{\mathbf{q}}_{2,n+1}^{\text{co}} = \mathbf{u}_{22} - \frac{\mathbf{q}_{2,n}^{\text{co}} - \mathbf{u}_{21}}{\Delta t}. \tag{23}$$

Substituting the update scheme into Eq. (12a), (12b) and the interface of Eq. (23) into Eq. (21) yields

$$\left(\alpha \Delta t + \beta + \frac{1}{\Delta t} \gamma\right) \left(\mathbf{u}_{22} - \frac{\mathbf{q}_{2,n}^{\text{co}} - \mathbf{u}_{21}}{\Delta t}\right) = \alpha (\mathbf{q}_{1,n+1}^{\text{co}} - \mathbf{q}_{2,n}^{\text{co}}) + \beta \dot{\mathbf{q}}_{1,n+1}^{\text{co}} + \frac{\gamma}{\Delta t} (\dot{\mathbf{q}}_{2,n}^{\text{co}} + \Delta t \ddot{\mathbf{q}}_{1,n+1}^{\text{co}}). \tag{24}$$

If inputs \mathbf{u}_{21} and \mathbf{u}_{22} can enforce Eq. (24), then the constraint in Eq. (21) will be satisfied. For simplicity, in the cosimulation, \mathbf{u}_{21} is given as

$$\mathbf{u}_{21} = \mathbf{q}_{2,n}^{\text{co}}, \tag{25}$$

which is substituted into Eq. (24) to generate the input \mathbf{u}_{22} as

$$\mathbf{u}_{22} = \frac{\Delta t}{\alpha \Delta t^2 + \beta \Delta t + \gamma} \left[\alpha \Delta t \frac{\mathbf{q}_{1,n+1}^{\text{co}} - \mathbf{q}_{2,n}^{\text{co}}}{\Delta t} + \beta \dot{\mathbf{q}}_{1,n+1}^{\text{co}} + \frac{1}{\Delta t} \gamma (\dot{\mathbf{q}}_{2,n}^{\text{co}} + \Delta t \ddot{\mathbf{q}}_{1,n+1}^{\text{co}}) \right]. \tag{26}$$

In Eq. (26), \mathbf{u}_{22} is the weighted average of three contributions: a generalized velocity vector obtained by the difference of $\mathbf{q}_{1,n+1}^{\text{co}}$ and $\mathbf{q}_{2,n}^{\text{co}}$, the velocity obtained from the smooth subsystem $\dot{\mathbf{q}}_{1,n+1}^{\text{co}}$, and the velocity vector obtained from the integration of $\ddot{\mathbf{q}}_{1,n+1}^{\text{co}}$.

2.3.3 A general coupling joint

When there are offsets between the coupling markers and the generalized coordinates of the coupling bodies, as shown in Fig. 1(b), the coupling constraints are formulated as functions of the markers' kinematic variables $\mathbf{q}_{1,A}^{\text{co}}$ and $\mathbf{q}_{2,B}^{\text{co}}$:

$$\mathbf{g}_{\text{co}}(\mathbf{q}_1^{\text{co}}, \mathbf{q}_2^{\text{co}}) = \mathbf{q}_{2,B}^{\text{co}} - \mathbf{q}_{1,A}^{\text{co}} = \mathbf{0}. \tag{27}$$

In $\mathbf{q}_{1,A}^{\text{co}}$ and $\mathbf{q}_{2,B}^{\text{co}}$ the displacement coordinate vectors $\mathbf{r}_{1,A}^{\text{co}}$ and $\mathbf{r}_{2,B}^{\text{co}}$ are obtained by

$$\mathbf{r}_{1,A}^{\text{co}} = \mathbf{r}_1^{\text{co}} + \mathbf{R}_1^{\text{co}} \mathbf{r}_{A,\text{loc}}, \tag{28a}$$

$$\mathbf{r}_{2,B}^{\text{co}} = \mathbf{r}_2^{\text{co}} + \mathbf{R}_2^{\text{co}} \mathbf{r}_{B,\text{loc}}, \tag{28b}$$

where $\mathbf{r}_{A,\text{loc}}$ and $\mathbf{r}_{B,\text{loc}}$ represent the offsets of the markers in the local reference frame of the bodies, \mathbf{r}_1^{co} and \mathbf{r}_2^{co} are the displacement coordinate vectors of the coupling bodies, $\mathbf{r}_{1R}^{\text{co}}$ and $\mathbf{r}_{2R}^{\text{co}}$ denote the rotation coordinates of the coupling bodies, and their corresponding rotation matrices are written as \mathbf{R}_1^{co} and \mathbf{R}_2^{co} , respectively. In $\mathbf{q}_{1,A}^{\text{co}}$ and $\mathbf{q}_{2,B}^{\text{co}}$ the rotation coordinates $\mathbf{r}_{1,AR}$ and $\mathbf{r}_{2,BR}$ are the same as those in $\mathbf{q}_{1,A}^{\text{co}}$ and $\mathbf{q}_{2,B}^{\text{co}}$:

$$\mathbf{r}_{1,AR}^{\text{co}} = \mathbf{r}_{1R}^{\text{co}}, \tag{29a}$$

$$\mathbf{r}_{2,BR}^{\text{co}} = \mathbf{r}_{2R}^{\text{co}}. \tag{29b}$$

From Eqs. (27)–(29b) the partial derivative $\mathbf{g}_{\text{co}/\mathbf{q}_1^{\text{co}}}$ can be obtained as

$$\mathbf{g}_{\text{co}/\mathbf{q}_1^{\text{co}}} = - \begin{bmatrix} \mathbf{E}_3, & -(\mathbf{R}_1^{\text{co}} \mathbf{r}_{A,loc}) \times \\ \mathbf{0}, & \mathbf{E}_3 \end{bmatrix}, \tag{30}$$

where \times represents the cross-product operator, and for a vector $\mathbf{a} \in \mathbb{R}^3$, $\mathbf{a} \times$ is the corresponding skew-symmetric matrix.³

In this case the original and modified input formats can be respectively expressed as functions of the coordinates of the markers, $\mathbf{q}_{1,A}^{\text{co}}$ and $\mathbf{q}_{2,B}^{\text{co}}$:

$$\mathbf{u}_{21} = \mathbf{q}_{1,A,n}^{\text{co}}, \tag{31a}$$

$$\mathbf{u}_{22} = \dot{\mathbf{q}}_{1,A,n+1}^{\text{co}}, \tag{31b}$$

and

$$\mathbf{u}_{21} = \mathbf{q}_{2,B,n}^{\text{co}}, \tag{32a}$$

$$\mathbf{u}_{22} = \frac{\Delta t}{\alpha \Delta t^2 + \beta \Delta t + \gamma} \left[\alpha \Delta t \frac{\mathbf{q}_{1,A,n+1}^{\text{co}} - \mathbf{q}_{2,B,n}^{\text{co}}}{\Delta t} + \beta \dot{\mathbf{q}}_{1,A,n+1}^{\text{co}} + \frac{1}{\Delta t} \gamma (\dot{\mathbf{q}}_{2,B,n}^{\text{co}} + \Delta t \ddot{\mathbf{q}}_{1,A,n+1}^{\text{co}}) \right]. \tag{32b}$$

We call the latter set of inputs “modified inputs.” Using these modified inputs, Baumgarte’s constraint stabilization in Eq. (21) is achieved during cosimulation, improving the algorithmic stability of the explicit cosimulation.

For a general joint that does not enforce all components of position and rotation, the reference is defined at Marker B to describe whether rotation or displacement along each axis in the local frame is allowed.

3 Cosimulation strategy

3.1 Cosimulation schemes

Cosimulation is performed by solving the subsystems separately, exchanging information at discrete time points. The time interval between these points is defined as a macrotime step. Although each subsystem can perform multiple time steps between macrotime steps, which is called multirate cosimulations, in this work, we assume the macrotime step and the time step in each subsystem to be the same. Explicit cosimulation schemes (in the sense of Hafner and Popper [43]), also called noniterative schemes [43], are established between the smooth and nonsmooth subsystems.

Explicit cosimulation can be structured according to two schemes:

³The operator that transforms the generic vector $\mathbf{a} \in \mathbb{R}^3$ into the skew-symmetric matrix $\mathbf{a} \times$ which in turn, when multiplies from the left another generic vector $\mathbf{b} \in \mathbb{R}^3$, yields the cross-product of \mathbf{a} and \mathbf{b} , namely $(\mathbf{a} \times) \mathbf{b} = \mathbf{a} \times \mathbf{b}$.

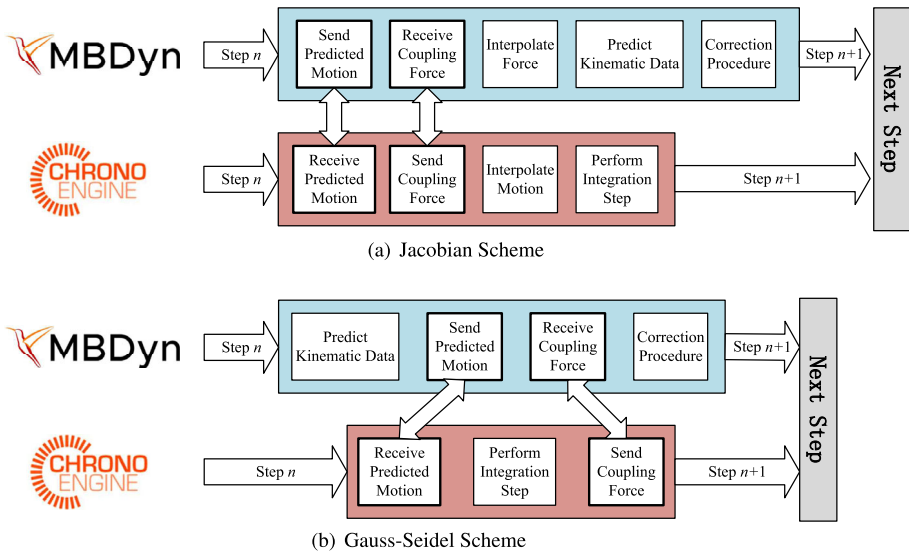


Fig. 2 Flowchart of different explicit cosimulation schemes for the smooth/nonsmooth cosimulation

- the Jacobi-type scheme [43], as shown in Fig. 2(a), in which all subsystems are simulated in parallel using extrapolated inputs for each time step;
- the Gauss–Seidel scheme, as shown in Fig. 2(b), in which subsystems are integrated serially at each time step: extrapolation is used to predict the input for the subsystem that is solved first; subsequently, it generates the input for the other subsystem, which is integrated afterward.

Both cosimulation algorithms can be applied to the cosimulation process using the modified inputs described in Sect. 2. It is worth mentioning that when the explicit Gauss–Seidel approach is employed, the cosimulation suffers from the risk of lacking zero-stability [27, 44], which depends on the mass ratios that contribute to the coupling kinematic constraints and cannot be guaranteed simply by using the modified inputs. Nonetheless, owing to its simplicity, we still use the explicit Gauss–Seidel cosimulation in the following examples to illustrate the procedure with modified inputs.

Assuming that the macrotime step is equal to the micro one, an explicit cosimulation scheme between the smooth and nonsmooth subsystems is depicted in Fig. 2(b). The nonsmooth subsystem is simulated first, based on the prescribed interface motion predicted by the smooth subsystem. Subsequently, the resulting reaction forces/moments are sent to the smooth subsystem. The choice of this sequence of operations naturally stemmed from the consideration that in the smooth subsystem the solution of the discrete equations (5a) and (5b) already predicts the solution to provide initial values for the Newton iteration used to solve the resulting nonlinear problem. Such initial values are obtained by extrapolation from the solution at previous time steps according to the aforementioned multistep integration scheme. Using the readily available predicted motion as output avoids repeating the extrapolation process. Alternatively, a different extrapolation formulation can be used for the coupling motion to improve the stability of cosimulation [45]. However, this investigation is beyond the scope of the present work.

In the smooth subsystem the derivatives of the state variables are extrapolated by Hermian interpolation, yielding the predicted generalized velocity vector $\dot{\mathbf{q}}_{1,n+1}^{(0)}$ [34]:

$$\dot{\mathbf{q}}_{1,n+1}^{(0)} = -\frac{12}{\Delta t}(\mathbf{q}_{1,n} - \mathbf{q}_{1,n-1}) + 8\dot{\mathbf{q}}_{1,n} + 5\dot{\mathbf{q}}_{1,n-1}, \tag{33}$$

which is substituted into the integration scheme in Eq. (6) to produce the predicted generalized coordinate

$$\begin{aligned} \mathbf{q}_{1,n+1}^{(0)} &= \sum_{i=1,2} a_i \mathbf{q}_{n+1-i} + \Delta t \sum_{i=0,2} b_i \dot{\mathbf{q}}_{n+1-i} \\ &= \sum_{i=1,2} a_i \mathbf{q}_{n+1-i} + \Delta t \sum_{i=1,2} b_i \dot{\mathbf{q}}_{n+1-i} + \Delta t b_0 \left[-\frac{12}{\Delta t}(\mathbf{q}_n - \mathbf{q}_{n-1}) + 8\dot{\mathbf{q}}_n + 5\dot{\mathbf{q}}_{n-1} \right] \\ &= (a_1 - 12b_0) \mathbf{q}_n + (a_2 + 12b_0) \mathbf{q}_{n-1} + \Delta t (b_1 + 8b_0) \dot{\mathbf{q}}_n + \Delta t (b_2 + 5b_0) \dot{\mathbf{q}}_{n-1}. \end{aligned} \tag{34}$$

For accelerations, a constant extrapolation formula is used:

$$\ddot{\mathbf{q}}_{1,n+1}^{(0)} = \ddot{\mathbf{q}}_{1,n}. \tag{35}$$

According to the input formats, these predicted state variables are substituted into Eq. (31a), (31b) or Eq. (32a), (32b) to yield inputs for the nonsmooth subsystem. The nonsmooth system is then solved to generate the reaction forces and moments for the smooth subsystem.

3.2 Implementation

Cosimulation of the coupled smooth/nonsmooth subsystem is implemented between the free general-purpose multibody dynamics software MBDyn and the open-source multiphysics C++ library, Project Chrono. The smooth subsystem is modeled in MBDyn, and the nonsmooth subsystem is modeled with the Chrono library. To achieve cosimulation between these two subsystems, a stable two-way communication layer is necessary.

In this work, we establish dynamic linking between these two solvers, as shown in Fig. 3, where the nonsmooth subsystem is embedded in MBDyn as an external module, implementing a user-defined force element. In detail, a run-time loadable module named `module-chrono-interface` was implemented, which can be loaded during the execution of MBDyn. The module consists of three parts: i) a user-defined element called `ChronoInterface`, ii) a set of MBDyn-Chrono interface functions, and iii) the model of the nonsmooth subsystem, implemented using the C++ libraries provided by Chrono. The `ChronoInterface` element is derived from the generic user-defined element in MBDyn, which has access to the solution process of MBDyn.

The `ChronoInterface` element can obtain kinematic data of bodies from the MBDyn solver and apply reaction forces and moments to the bodies themselves. Additionally, to gather the reaction forces and moments, this element also sends commands to the nonsmooth subsystem, such as requesting the simulation of the nonsmooth subsystem by calling the related function and reading/writing information related to the nonsmooth subsystem from/to files.

The second part of the module is a set of MBDyn-Chrono interface functions. These functions act as an intermediate communication layer: they receive kinematic data and commands from the `ChronoInterface` element and pass them to the nonsmooth subsystem,

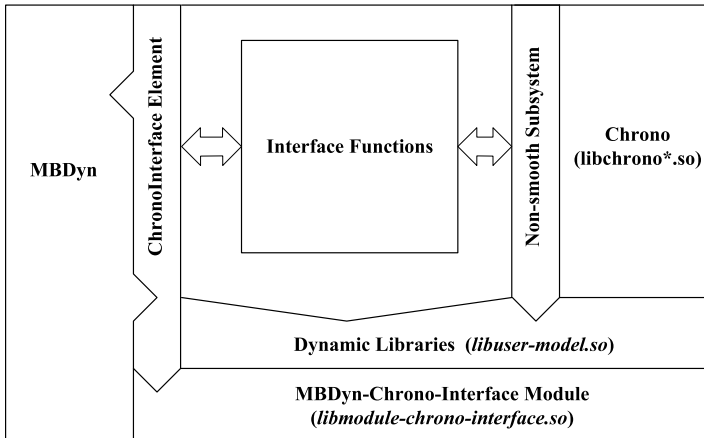


Fig. 3 Schematics of communication layer between MBDyn and Chrono

receive dynamic data and messages from the nonsmooth subsystem, and hand them over to the ChronoInterface element.

The third part of the run-time loadable module is the nonsmooth subsystem, which needs to be modeled by the user in the function named `MBDyn_CE_CEModel_Create()`. In the process of modeling, the physical components of the nonsmooth subsystem are provided by the Chrono library.

According to the dependencies of the different parts, the interface functions and the user-defined nonsmooth subsystem are compiled together with the dynamic libraries in Chrono to generate a dynamic library for the ChronoInterface element called `libuser-model.so`. With the generated library, the ChronoInterface element generates a new dynamic library, called `libchrono-interface-module.so`, which is run-time loaded in MBDyn, by defining the related statement in the input file. Overall, this communication layer provides a convenient way to establish cosimulation between MBDyn and Project Chrono, by which the cosimulation scheme in Fig. 2(b) is achieved.

The previously mentioned explicit Gauss–Seidel cosimulation scheme is similar to the first step of the implicit cosimulation scheme in [7]. The proposed explicit cosimulation scheme differs from that of [7] in the following three aspects: (1) apart from the cosimulation procedure illustrated in Fig. 2(b), the implicit cosimulation scheme includes an iterative process, by which the simulation of the nonsmooth subsystem(s) is embedded into the correction process of Eqs. (10a)–(10c) of the smooth solver; the cosimulation moves to the subsequent step only when the convergence of the iterative process is achieved; (2) the proposed explicit cosimulation scheme focuses on evaluating the ability of different types of inputs, consisting of position-, velocity-, and acceleration-level kinematic quantities and their combinations to obtain a stable and accurate cosimulation, whereas the implicit scheme only uses the position-level generalized physical coordinates during cosimulation; (3) last but not least, the proposed explicit cosimulation scheme is implemented in the form of a user-defined run-time loadable module embedded in MBDyn, `module-chrono-interface`, by which the aforementioned cosimulation schemes, such as the Jacobi- or Gauss–Seidel-type explicit or implicit cosimulations using the modified inputs, are all achieved in the element interface, `ChronoInterface`, whereas the data exchange approach used for the implicit scheme illustrated in [7] was built on interprocess communi-

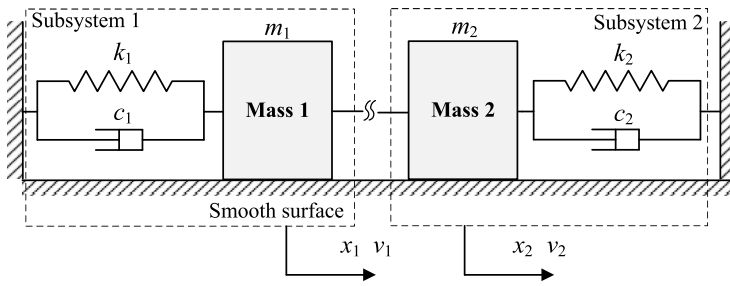


Fig. 4 Decomposition of the two-mass oscillator

ication through Unix sockets, the cosimulation strategies being defined in the specific subsystems. This may appear an implementation detail but represents a substantial generalization and a great simplification in the use of the proposed cosimulation setup.

4 Numerical experiments

4.1 Linear two-mass oscillator

The linear two-mass oscillator of Fig. 4 is commonly used to test the performance of cosimulation methods in the literature [21]. Also in this work, it is used to test the proposed cosimulation configuration and to show the influence of the modified inputs on its algorithmic stability. In the tests the recurrence equations of the cosimulation are deduced to produce the stability plots with different inputs and to show the influence of the modified inputs. It is worth mentioning that although the two-mass oscillator is a smooth system, it is still decomposed into two subsystems, and one of them is integrated using the method proposed for nonsmooth subsystems.

4.1.1 Recurrence scheme

According to the methods in Sect. 2, the two-mass oscillator with constraints in Fig. 4 is decomposed into two subsystems:

$$\begin{cases} \dot{x}_1 = v_1, \\ \dot{v}_1 = -\frac{k_1}{m_1}x_1 - \frac{c_1}{m_1}v_1 + \frac{1}{m_1}\lambda, \end{cases} \quad (36a)$$

$$\begin{cases} \dot{x}_2 = v_2, \\ \dot{v}_2 = -\frac{k_2}{m_2}x_2 - \frac{c_2}{m_2}v_2 - \frac{1}{m_2}\lambda, \\ x_2 - x_1 = 0, \end{cases} \quad (36b)$$

where m_i , k_i , and c_i ($i = 1, 2$) are the mass, stiffness, and damping coefficients for bodies 1 and 2, as shown in Fig. 4, x_i and v_i ($i = 1, 2$) are the position and velocity of the two bodies, and λ is the Lagrange multiplier corresponding to the algebraic constraint.

Equations (36a) are formulated using the method for the smooth subsystem and are integrated by the implicit linear two-step scheme in Eq. (6). Substituting Eq. (36a) into the

integration scheme (6) yields

$$x_{1,n+1} = \sum_{j=1,2} a_j x_{1,n+1-j} + \Delta t \sum_{j=1,2} b_j v_{1,n+1-j} + \Delta t b_0 v_{1,n+1}, \quad (37a)$$

$$v_{1,n+1} = \sum_{j=1,2} a_j v_{1,n+1-j} + \Delta t \sum_{j=0,1,2} b_j \left(-\frac{k_1}{m_1} x_{1,n+1-j} - \frac{c_1}{m_1} v_{1,n+1-j} + \frac{1}{m_1} u_{1,n+1-j} \right), \quad (37b)$$

where u_1 is the input.

Equation (36a) and (36b) is formulated using the formulation proposed for the nonsmooth subsystem in Eq. (11a)–(11d), although the subsystem analyzed here is smooth, as

$$x_{2,n+1} = x_{2,n} + \Delta t v_{2,n+1}, \quad (38a)$$

$$v_{2,n+1} = v_{2,n} - \frac{\Delta t k_2}{m_2} x_{2,n} - \frac{\Delta t c_2}{m_2} v_{2,n} - \frac{\Delta t}{m_2} \lambda_{n+1}, \quad (38b)$$

$$0 = \frac{x_{2,n} - u_{21,n+1}}{\Delta t} + v_{2,n+1} - u_{22,n+1}, \quad (38c)$$

where u_{21} and u_{22} are inputs received from subsystem 1; the forces generated by the spring and damper are regarded as parts of the external forces \mathbf{f}_2 .

For the inputs in Eqs. (37a), (37b) and (38a)–(38c), if those that have been defined as the original inputs in Eq. (31a), (31b) are employed, then we obtain

$$u_{1,n+1} = \lambda_{n+1}, \quad (39a)$$

$$u_{21,n+1} = x_{1,n}, \quad (39b)$$

$$u_{22,n+1} = v_{1,n+1}^{(0)} = -\frac{12}{\Delta t} (x_{1,n} - x_{1,n-1}) + 8v_{1,n} + 5v_{1,n-1}, \quad (39c)$$

where $v_{1,n+1}^{(0)}$ is obtained using the extrapolation formula (33). Collecting Eqs. (37a)–(39c) yields the recurrence equations

$$\mathbf{A}_1 \mathbf{z}_{n+1} = \mathbf{A}_2 \mathbf{z}_n + \mathbf{A}_3 \mathbf{z}_{n-1}, \quad (40)$$

where

$$\mathbf{z}_{n+1} = [x_{1,n+1}, v_{1,n+1}, x_{2,n+1}, v_{2,n+1}, \lambda_{n+1}, u_{1,n+1}, u_{21,n+1}, u_{22,n+1}]^T, \quad (41)$$

and the coefficient matrices \mathbf{A}_i ($i = 1, 2, 3$) are related to the system parameters, the algorithmic parameters ρ_∞ , and time-step sizes Δt .

Alternatively, if “modified inputs” in Eq. (32a), (32b) are employed to incorporate Baumgarte’s stabilization method during the explicit cosimulation, then we obtain

$$u_{1,n+1} = \lambda_{n+1}, \quad (42a)$$

$$u_{21,n+1} = x_{2,n}, \quad (42b)$$

$$u_{22,n+1} = \frac{\Delta t}{\alpha \Delta t^2 + \beta \Delta t + \gamma} \left[\alpha \left(x_{1,n+1}^{(0)} - x_{2,n} \right) + \beta v_{1,n+1}^{(0)} + \frac{1}{\Delta t} \gamma \left(v_{2,n} + \Delta t \dot{v}_{1,n+1}^{(0)} \right) \right]$$

$$= \Theta_0 \left(\Theta_1 x_{1,n} + \Theta_2 x_{1,n-1} + \Theta_3 v_{1,n} + \Theta_4 v_{1,n-1} + \frac{\gamma}{m_1} u_{1,n} - \alpha x_{2,n} + \frac{\gamma}{\Delta t} v_{2,n} \right), \tag{42c}$$

where the superscript $(\cdot)^{(0)}$ denotes the extrapolated variables of Eqs. (33)–(35), and

$$\Theta_0 = \frac{\Delta t}{\alpha \Delta t^2 + \beta \Delta t + \gamma}, \tag{43a}$$

$$\Theta_1 = (a_1 - 12b_0)\alpha - \frac{12}{\Delta t}\beta - \frac{k_1}{m_1}\gamma, \tag{43b}$$

$$\Theta_2 = (a_2 + 12b_0)\alpha + \frac{12}{\Delta t}\beta, \tag{43c}$$

$$\Theta_3 = (\Delta t b_1 + 8\Delta t b_0)\alpha + 8\beta - \frac{c_1}{m_1}\gamma, \tag{43d}$$

$$\Theta_4 = (\Delta t b_2 + 5\Delta t b_0)\alpha + 5\beta. \tag{43e}$$

Similarly to the case that uses the original inputs, the recurrence equations can be obtained by combining Eqs. (36b), (37a), (37b) and (42a)–(42c) expressed in the form (40) with a new set of coefficient matrices \mathbf{A} ($i = 1, 2, 3$), which are also related to the parameters α , β , and γ in the modified inputs.

4.1.2 Stability results of the recurrence equations

From a physical point of view, the linear two-mass oscillator system is called asymptotically stable if its parameters satisfy the requirements $m_i > 0$, $k_i > 0$, $c_i > 0$. For an asymptotically stable two-mass oscillator, cosimulation results using the proposed explicit cosimulation scheme can be obtained from the solution of the recurrence equations (40). They can be solved using the exponential approach, as $\mathbf{z}_n = \hat{\mathbf{z}} \cdot \hat{\lambda}^n$, where $\hat{\lambda}$ is the generic eigenvalue, and $\hat{\mathbf{z}}$ is the corresponding eigenvector of the recurrence equation [46]. The stability of the solution depends on the spectral radius $\tilde{\rho} = \max(|\hat{\lambda}_j|)$ of the recurrence equation, where $\hat{\lambda}_j$ is the j th eigenvalue of the latter. When $\tilde{\rho} > 1$, the recurrence equation generates an unstable solution, and the corresponding cosimulation is (algorithmically) unstable [46]. Otherwise, when $\tilde{\rho} \leq 1$, the cosimulation is (algorithmically) stable [46].

In the tests, we consider a set of parameters, where $m_1 = 1.0$ kg, and other system parameters are defined as functions related to equivalent natural frequencies ω_{0i} ($i = 1, 2$) and damping factors ξ_i of the subsystems:

$$\begin{cases} k_i = m_i \omega_{0i}^2 \text{ N m}^{-1}, \\ c_i = 2m_i \omega_{0i} \xi_i \text{ N s m}^{-1}, \end{cases} \tag{44}$$

where

$$m_2 = 1.0 \text{ kg}, \quad \omega_{02} = 0.0 \text{ rad s}^{-1}, \quad \xi_1 = \xi_2 = 0.0. \tag{45}$$

Different sets of parameters are employed for the modified inputs, namely:

$$\alpha = \frac{2}{\Delta t^2}, \quad \beta = \frac{2}{\Delta t}, \quad \gamma = 1, \tag{46a}$$

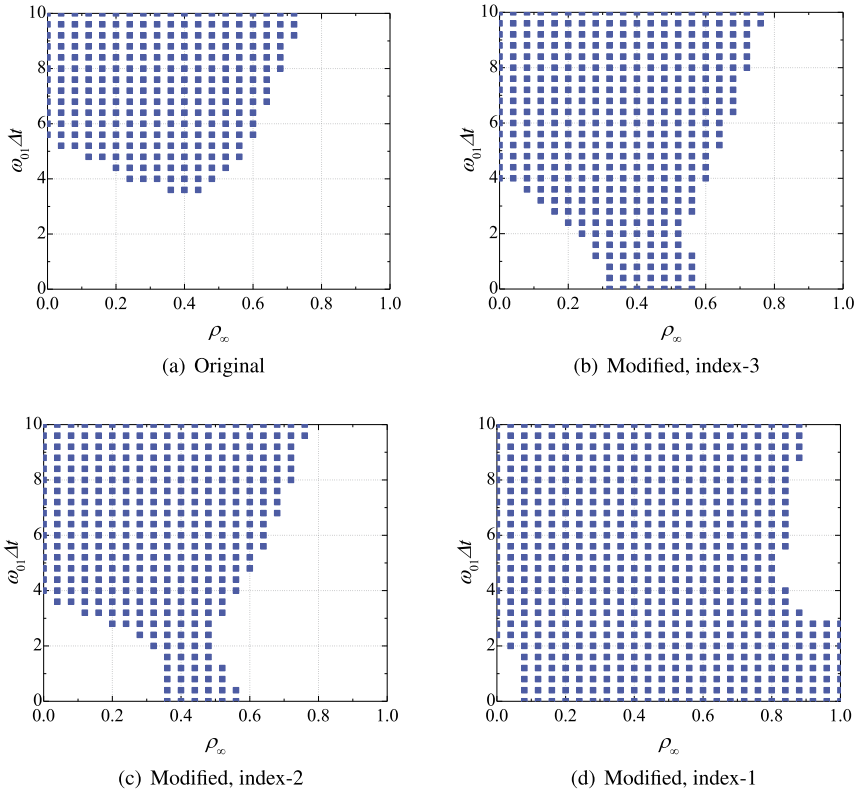


Fig. 5 Stability plots of cosimulation schemes with: **(a)** the original inputs of Eq. (20); **(b)** index-3: $\alpha = 1$, $\beta = 0$, and $\gamma = 0$; **(c)** index-2: $\alpha = 1/\Delta t^2$, $\beta = 1/\Delta t$, and $\gamma = 0$; **(d)** index-1: $\alpha = 2/\Delta t^2$, $\beta = 2/\Delta t$, and $\gamma = 1$

$$\alpha = \frac{1}{\Delta t}, \quad \beta = 1, \quad \gamma = 0, \quad (46b)$$

$$\alpha = 1, \quad \beta = 0, \quad \gamma = 0, \quad (46c)$$

which correspond to index-1, index-2, and index-3 monolithic DAEs, respectively. Two-dimensional stability plots as functions of $\omega_{01} \Delta t$ and ρ_∞ are shown in Fig. 5 with the original and modified inputs. In the figure the points indicate that $\tilde{\rho} \leq 1$ is satisfied using the corresponding set of $\omega_{01} \Delta t$ and ρ_∞ , i.e., stable cosimulation results are obtained. We can see that for a simple, undamped case, the cosimulation using the original inputs cannot give stable results for small values of $\omega_{01} \Delta t$, at least for a broad range of ρ_∞ . With the same system parameters in Eq. (45), this situation can be improved using the modified inputs, which can always produce stable solutions with a small $\omega_{01} \Delta t$, provided that a suitable ρ_∞ is employed in the smooth multistep integrator. Apart from this, Fig. 5(d) also shows that the modified inputs with the index-1 format, corresponding to the parameters in Eq. (46a), possess the largest stability region.

Different input parameters are also employed to test the stability of the explicit cosimulation scheme, as shown in Fig. 6, where the stability plots of an implicit cosimulation scheme developed in [7] are also provided as a reference. We can see that when α and β

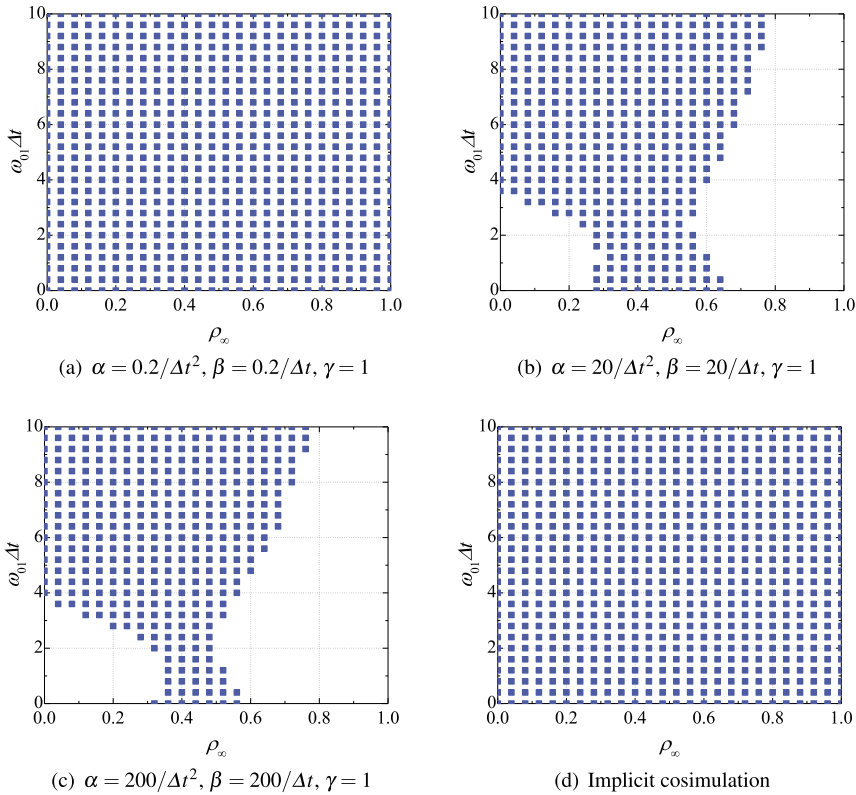


Fig. 6 Stability plots of cosimulation schemes with (a) $\alpha = 0.2/\Delta t^2$, $\beta = 0.2/\Delta t$, and $\gamma = 1$, (b) $\alpha = 20/\Delta t^2$, $\beta = 20/\Delta t$, and $\gamma = 1$, (c) $\alpha = 200/\Delta t^2$, $\beta = 200/\Delta t$, and $\gamma = 1$, and (d) implicit cosimulation scheme

are smaller than those used in Eq. (46a), the explicit cosimulation scheme can offer stable results in all $\omega_{01}\Delta t$ and ρ_∞ tested combinations close to the performance of the implicit cosimulation scheme. When larger α and β are used, the stability results are similar to those of the explicit cosimulation scheme with the input parameters of Eq. (46b), since $\alpha \gg \gamma$ and $\beta \gg \gamma$ imply that the constraints error $\gamma (\dot{v}_{2,n+1} - \dot{v}_{1,n+1}^{(0)})$ can be neglected, compared with that of the other two contributions.

Stability plots with different mass ratios are also shown in Fig. 7. We can notice that a large mass ratio ($m_2 \gg m_1$) may spoil the zero-stability of the explicit Gauss–Seidel cosimulation, regardless of whether the original or the index-1 modified inputs are used, although the cosimulation using the index-1 modified inputs shows better stability properties. The influence of the mass ratio is consistent with analogous conclusions in the literature that cosimulation may lack zero-stability when the mass ratio is large [27, 44].

4.1.3 Simulation results

With initial position $x_1 = x_2 = 0.0$ m and initial velocity $v_1 = v_2 = 0.1$ m/s, the explicit cosimulation between MBDyn and Chrono is performed with $\rho_\infty = 0.56$ and $\omega_{01}^2 = 5000$ s⁻². The cosimulation results with original and modified inputs of Eqs. (46a)–(46c) are shown

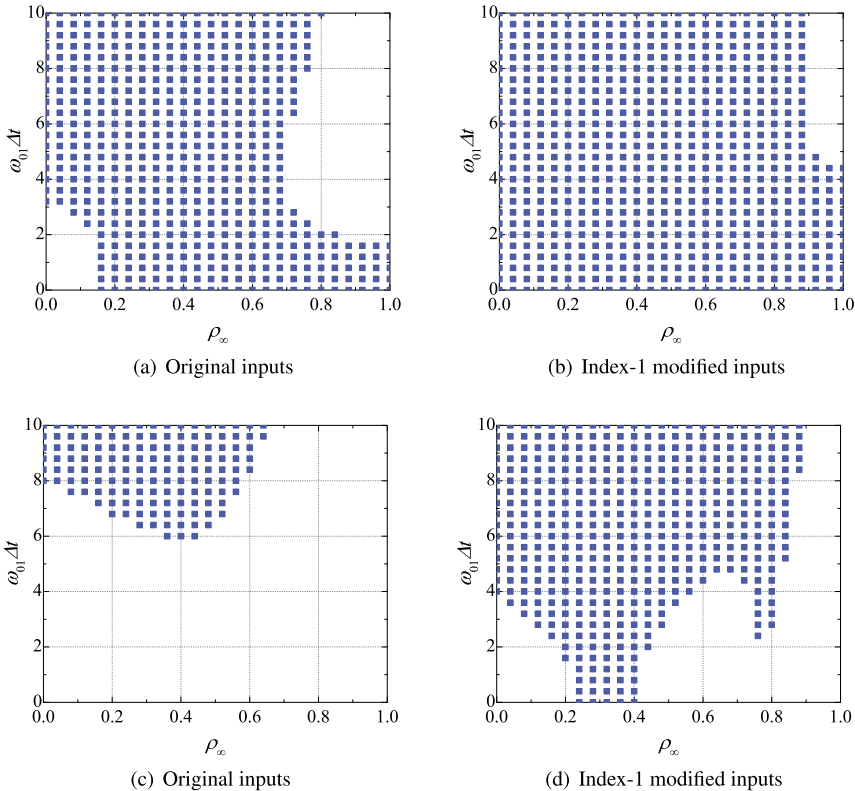


Fig. 7 Stability plots of cosimulation schemes with different mass ratios: **(a)** original inputs and $m_2/m_1 = 0.5$, **(b)** index-1 modified inputs and $m_2/m_1 = 0.5$, **(c)** original inputs and $m_2/m_1 = 2.0$, **(d)** index-1 modified inputs and $m_2/m_1 = 2.0$

in Fig. 8, where the results obtained by a monolithic simulation in MBDyn are shown as dotted lines for reference. The cosimulation results are consistent with those obtained from the previously mentioned recurrence equations (40). In detail, the solution using the original inputs is unstable because the corresponding spectral radius $\tilde{\rho} \approx 1.11 > 1$, whereas algorithmically stable results are obtained with the other three sets of inputs, since the stability condition $\tilde{\rho} \approx 0.999 \leq 1$ is satisfied. However, Fig. 8 also shows a clear difference between the results obtained with monolithic and explicit cosimulations, owing to the algorithmic dissipation introduced by the explicit coupling. Such difference can be reduced using a smaller time step size, as shown in Fig. 9, where the relative errors of the explicit cosimulation of the interval $t \in [0, 0.5]$ s using different step sizes are given. The relative errors of the implicit cosimulation scheme proposed in [7] (a maximum of 10 iterations is allowed, and the co-simulation tolerance is 1×10^{-6}) are also plotted as a reference. In the figure the relative errors are computed using the L^1 norm [7]:

$$\|\text{error}\|_1 = \frac{\sum_{n=0}^N |x_n - x(t_n)|}{\sum_{n=0}^N |x(t_n)|}, \quad (47)$$

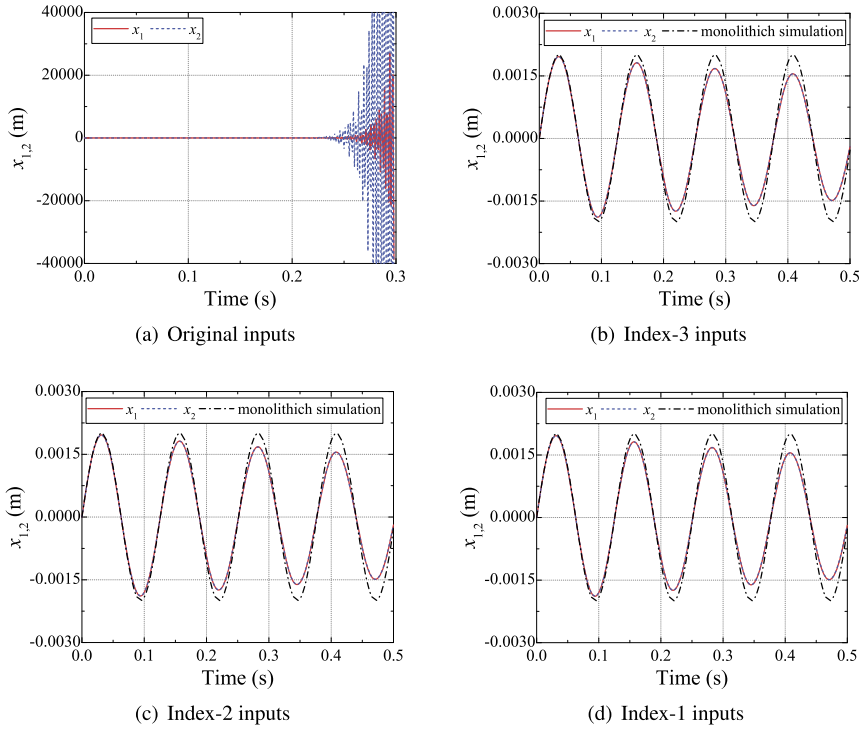


Fig. 8 Time histories of x_i using the proposed explicit cosimulation schemes: (a) using original inputs, (b) using index-3 parameters of Eq. (46c), (c) using index-2 parameters of Eq. (46b), and (d) using index-1 parameters of Eq. (46a)

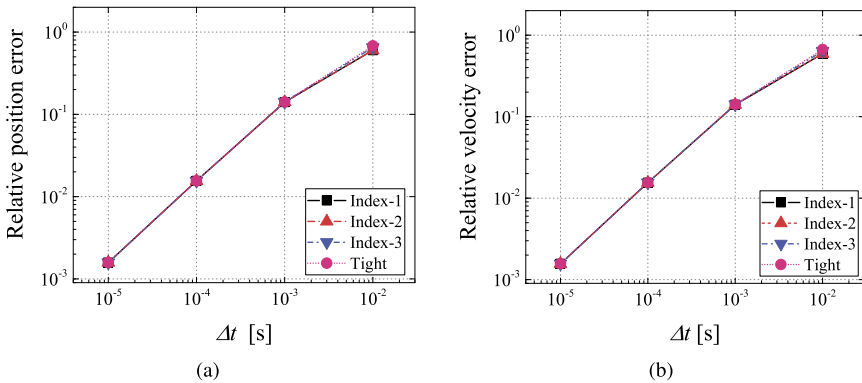
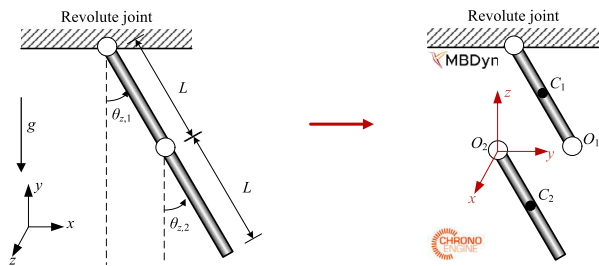


Fig. 9 Relative errors of the developed explicit co-simulation schemes: (a) relative position error; (b) relative velocity error

where x_n is the numerical solution at time t_n , and $x(t_n)$ is the theoretical solution at time t_n . The explicit cosimulation scheme using modified inputs approximately has the first-order accuracy as the implicit one. Table 1 reports the CPU times required by the different co-

Table 1 CPU time (s) of different cosimulation schemes

Co-simulation schemes	$\Delta t = 10^{-2}$ s	$\Delta t = 10^{-3}$ s	$\Delta t = 10^{-4}$ s	$\Delta t = 10^{-5}$ s
Explicit cosimulation (index-1)	0.11	1.20	11.81	125.22
Explicit cosimulation (index-2)	0.11	1.22	11.55	121.56
Explicit cosimulation (index-3)	0.10	1.16	11.52	122.40
Implicit cosimulation	1.18	13.39	117.77	1012.66

Fig. 10 Decomposition of a double-pendulum system

simulation schemes; an Intel i5-8300H CPU @ 2.30 GHz with 8G RAM is employed using only one thread. The table shows that the proposed explicit cosimulation scheme needs much less computation time than the implicit cosimulation one while offering comparable accuracy.

In this subsection, we employ the two-mass oscillator with kinematic constraints to assess the stability of the cosimulation with different input formats. Stability plots are presented to show the effects of the modified inputs. The modified inputs can enlarge the stability region; using index-1 modified inputs, we obtain the largest stability region, although it still cannot guarantee the zero-stability of the explicit Gauss–Seidel cosimulation. This conclusion is verified by cosimulation results between MBDyn and Chrono, which also show that the original input format leads to unstable results. Besides, compared to the implicit cosimulation scheme [7], the explicit one is more efficient as we would expect, as it requires much less computational time to achieve comparable accuracy.

4.2 Double-pendulum system

Two bodies, one connected to the ground and both connected to each other by revolute joints, form a double pendulum system. For cosimulation, the system is decomposed at the revolute joint $O_1 - O_2$ that connects the two bodies as shown in Fig. 10. One pendulum is modeled in MBDyn; the coordinates of point C_1 are employed as generalized coordinates, and the motion of O_1 is generated for another subsystem. In the other subsystem the displacements and rotations of point C_2 are employed as generalized coordinates, and the coupling constraints are imposed to point O_2 . A reference frame at O_2 , rigidly connected to the pendulum, is also established to describe whether the constraints are active, as shown in Fig. 10. In this reference, a revolute joint is set up by allowing rotation about axis x , whereas rotation about the other two axes and displacement along all three axes are constrained.

The system parameters are

$$L = 0.4 \text{ m}, \quad m_1 = m_2 = 1 \text{ kg}, \quad \mathbf{J}_{C_1} = \mathbf{J}_{C_2} = \mathbf{E}_3 \text{ kg m}^2, \quad \theta_{z,i} = \frac{\pi}{2}, \quad \dot{\theta}_{z,i} = 0.5 \text{ rad s}^{-1}, \quad (48)$$

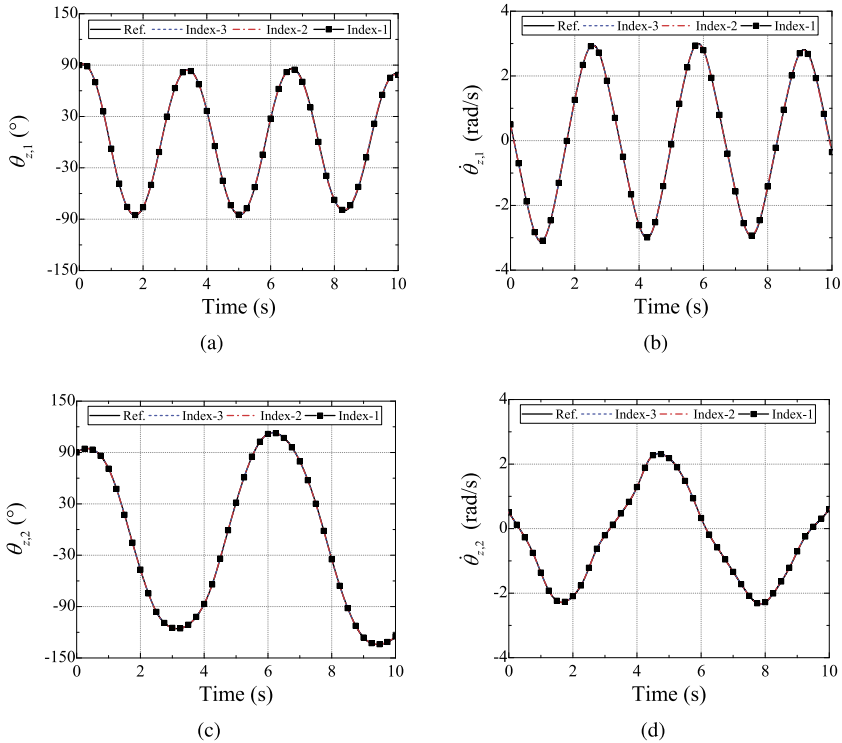


Fig. 11 Time histories of $\theta_{z,i}$ and $\dot{\theta}_{z,i}$ using the developed explicit cosimulation schemes: (a) $\theta_{z,1}$, (b) $\dot{\theta}_{z,1}$, (c) $\theta_{z,2}$, and (d) $\dot{\theta}_{z,2}$

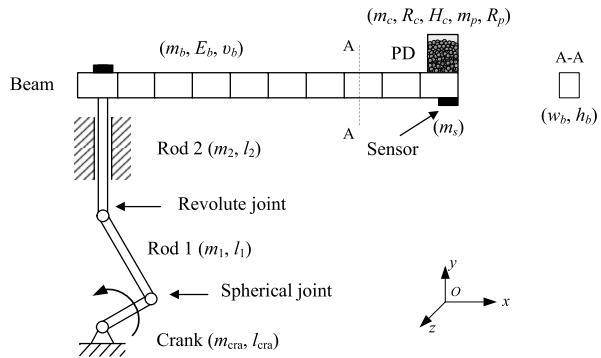
where L and m_i ($i = 1, 2$) are the length and mass of the bodies, \mathbf{J}_{C_i} represents the moment of inertia with respect to point C_i , $\mathbf{E}_3 \in \mathbb{R}^{3 \times 3}$ is the identity matrix, $\theta_{z,i}$ and $\dot{\theta}_{z,i}$ denote the initial rotation and angular velocity about axis z in the global reference. Using these parameters, the system is cosimulated by the proposed explicit cosimulation scheme with index-1, index-2, and index-3 parameters in Eqs. (46a)–(46c). The time histories of $\theta_{z,i}$ and $\dot{\theta}_{z,i}$ are shown in Fig. 11, where the monolithic simulation results obtained using MBDyn are plotted as references. The cosimulation results agree well with those obtained using the monolithic simulation. This example demonstrates that the developed explicit cosimulation scheme can effectively deal with coupled systems connected by a coupling revolute joint.

4.3 Smooth system coupled with a nonsmooth particle damper

4.3.1 System description

An experimental setup used to test the amount of damping provided by a particle damper (PD) is modeled and cosimulated in this section. The system shown in Fig. 12 consists of a crank-slider mechanism, a beam, a particle damper, and a sensor. One tip of the beam is connected to the slider of the slider-crank mechanism, which forces the transverse displacement of the beam. The PD is mounted at the other tip of the beam to limit its vibrations. A sensor is also attached at the location of the PD.

Fig. 12 Diagram of the smooth/nonsmooth coupled system with a particle damper



To numerically study the dynamical behavior of the setup in Fig. 12, a three-dimensional model is built. The system is decomposed into two subsystems at the constraints connecting the beam and the PD. After decomposition, the smooth subsystem is modeled in MBDyn, which consists of a crank-slider mechanism, a beam, and a sensor, whereas the nonsmooth subsystem consisting of a PD is modeled using the Chrono library.

In the smooth subsystem the crank-slider mechanism is modeled as three bodies: the crank, the connecting rod (rod 1), and the slider (rod 2), as shown in Fig. 12. A rheonomic constraint is imposed on the crank to enforce its rotation at constant angular velocity ω_{cra} , so that the crank-slider mechanism can drive the beam's constrained end to move along axis y . To avoid constraint redundancy, the joint connecting the crank and rod 1 is modeled as a spherical hinge, and the joint between rod 1 and rod 2 is modeled as a revolute joint. Rod 2 is also constrained by a translation joint, which forces the rod to move along the guide. Besides, rod 2 is rigidly connected to the beam at one tip to excite the beam's transverse displacement.

The beam is modeled in MBDyn using the original geometrically exact finite volume approach [47, 48]. It is meshed using 10 three-node beam elements for a total of 21 nodes. A linear elastic constitutive law is considered with no structural damping. A PD is rigidly connected to the other end of the beam. The connection is represented by a joint that imposes all components of relative displacement and rotation.

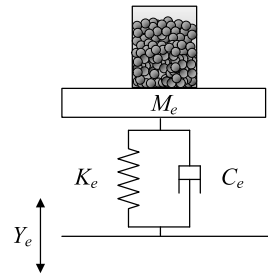
The particles are treated as rigid three-dimensional spheres of identical size. Among the particles, nonsmooth interactions are modeled as frictional unilateral contacts using the nonsmooth formulation [29], which are solved using the Chrono library. The particles are collected in a cylindrical container and modeled as a rigid body. Its outer height is H_c . Its ceiling can be adjusted for different inner heights $H_{c_{in}}$, although only one is considered in the present work. Impacts and friction between particles and the container are described using unilateral frictional constraints. In the simulation the same dynamic and static friction coefficients μ_p are assumed for interparticle and particle–container interaction. Fully plastic impacts are considered.

The parameters of each body are indicated by the corresponding subscripts, as shown in Fig. 12, where $(\cdot)_{\text{cra}}$, $(\cdot)_1$, $(\cdot)_2$, $(\cdot)_b$, $(\cdot)_c$, $(\cdot)_p$, and $(\cdot)_s$, respectively, represent the crank, rod 1, rod 2, the beam, the container of the PD, the particles, and the sensor. The mass, length, and density of a body are denoted by m , l , and ρ , where the mass of the particles refers to the total mass of all particles; R_r is the radius of the crank, rod 1, and rod 2; E_b and ν_b are Young's modulus and Poisson's ratio of the beam; w_b and h_b are the width and height of the beam cross-section; R_c and H_c are the base radius and the height of the container; R_p is the particle radius; J_{sx} , J_{sy} , J_{sz} , J_{cx} , J_{cy} , and J_{cz} are the diagonal elements of the inertia tensor

Table 2 Parameters for the smooth/nonsmooth coupled system with particle damper

	Parameters	Value	Parameters	Value	Parameters	Value
Crank-Slider	l_{cra} (mm)	0.25	l_1 (mm)	10.0	l_2 (mm)	10.0
	m_{cra} (kg)	1.0	m_1 (kg)	1.0	m_2 (kg)	2.0
	R_r (mm)	0.5				
Beam/Sensor	E_b (GPa)	210	v_b	0.33	m_b (g)	117.7
	ρ_b (kg/m ³)	7845	h_b (mm)	2.0	w_b (mm)	30.0
	l_b (mm)	250.0	m_s (g)	12.7	J_{sx} (kg mm ²)	1.05833
	J_{sy} (kg mm ²)	1.27	J_{sz} (kg mm ²)	1.05833		
PD	m_c (g)	35.8	R_c (mm)	10.0	H_c (mm)	20.0
	μ_p	0.52	ρ_p (kg/m ³)	11344	m_p (g)	7.0
	R_p (mm)	1.0	J_{cx} (kg mm ²)	2.98333	J_{cy} (kg mm ²)	3.58
	J_{cz} (kg mm ²)	2.98333				

Fig. 13 Simplified system with a particle damper



of the bodies lumped at the free end of the beam, the subscripts s and c referring to the sensor and the particles’ container, respectively. The parameter values used in the analysis are listed in Table 2.

As a reference, a simplified model is also simulated, as shown in Fig. 13, where the smooth subsystem is described by an equivalent single degree of freedom system, and the excitation Y_e is obtained from the motion of the point that connects rod 2 and the beam. The equivalent mass and stiffness are obtained from [49] as

$$M_e = 0.24m_b + m_{attach}, \quad K_e = \frac{3E_b I_{bz}}{l_b^3}, \quad (49)$$

where m_{attach} denotes the mass attached to the beam, including that of the sensor; $I_{bz} = (w_b h_b^3)/12$ is the moment of inertia of the beam cross-section. Since the beam is modeled with no structural damping, the equivalent damping ratio C_e is zero. Using the parameters in Table 2, $M_e = 0.076742$ kg and $K_e = 806.4$ N/m are obtained.

4.3.2 Simulation results

In the numerical experiments the proposed explicit cosimulation scheme between the smooth method in MBDyn and the nonsmooth method using the Chrono library is employed for solving the coupled system with a PD in Fig. 12. To obtain algorithmically stable numerical results, the modified inputs are used setting the coefficients α , β , and γ as in Eq. (22).

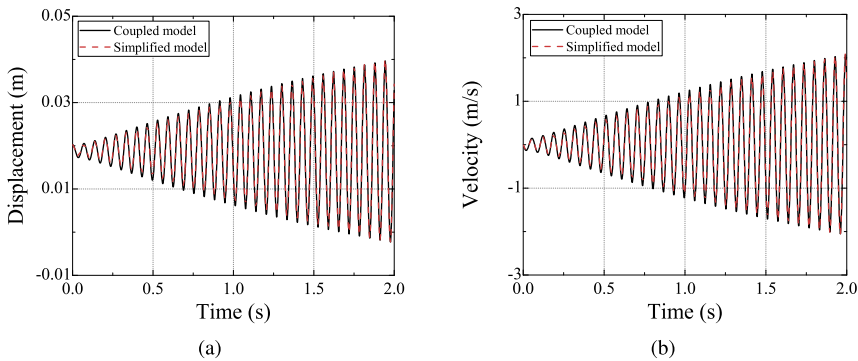


Fig. 14 Time histories of (a) displacement and (b) velocity, along axis y of the container without particles

The simplified system with a particle damper of Fig. 13 is simulated monolithically using the Chrono library to verify the results of the explicit cosimulation.

Initially, the crank-slider mechanism is set at the highest position, the crank and rod 1 being aligned with axis y . The initial angular velocity of the crank is $\omega_{\text{cra}} = 31.0\pi$ rad/s. A particle damper with inner height $H_{\text{cin}} = 11.84$ mm is used, which is able to give satisfactory damping for vibrations around the first natural frequency of the system, between 15 Hz and 16 Hz [7]. Considering the configuration of the crank-slider mechanism and its initial conditions, the excitation Y_e in the simplified model can be expressed as

$$Y_e = \sqrt{l_1^2 - l_{\text{cra}}^2 \cos^2(\omega_{\text{cra}}t + 0.5\pi)} + l_{\text{cra}} \sin(\omega_{\text{cra}}t + 0.5\pi) + l_2, \quad (50)$$

where t denotes the simulation time. Time step sizes for the explicit cosimulation scheme and the monolithic one are $\Delta t = 10^{-4}$ s.

A case without particles is studied first, where the total mass of the particles is treated as a lumped mass attached to the beam's end. In this case the natural frequencies of the beam with the particle mass lumped at the tip can be obtained by modal analysis [7], yielding the first three natural frequencies 15.619 Hz, 127.460 Hz, and 381.082 Hz. In the numerical experiments, since the angular velocity is set as $\omega_{\text{cra}} = 31.0\pi$ rad/s, close to the first natural frequency, resonance is observed in the results, as shown in Fig. 14, where the y component of displacement and velocity of the container are plotted. In the figure the solid black lines represent the results obtained with the coupled model using the explicit cosimulation scheme, and the red dashed lines show the results obtained with the monolithic simulation of the simplified model. These lines almost overlap, which further verifies the effectiveness of the explicit cosimulation scheme.

In the second case, we consider a particle damper filled with particles. The total mass of the particles and the properties of particles are listed in Table 2. All other conditions are the same as in the first case. The displacements and velocities of the container along axis y are plotted in Fig. 15, where the results obtained from the simplified model without particles are also shown. We can see that the explicit cosimulation scheme predicts satisfactory results, which are very close to the monolithic simulation results. All these results indicate that the PD can effectively suppress the resonance of the beam when the excitation frequency is close to the first natural frequency of the beam. Besides, because of the chaotic motion of the particles, the symmetry of the problem is broken, and the beam twists, as shown in Fig. 16, where the rotation about axis x and the torsional moment between the PD and the

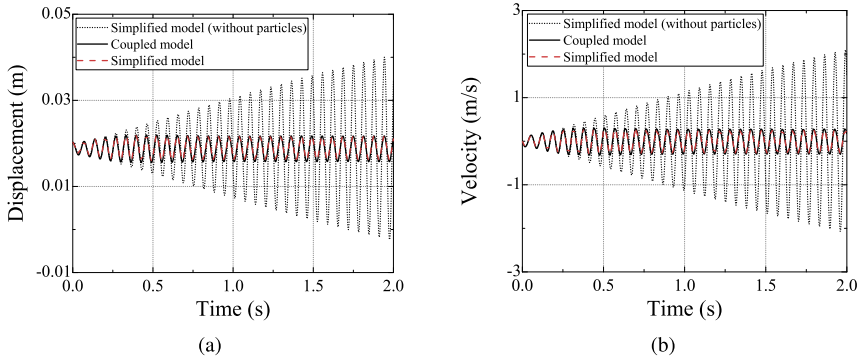


Fig. 15 Time histories of (a) displacement and (b) velocity along axis y of the container

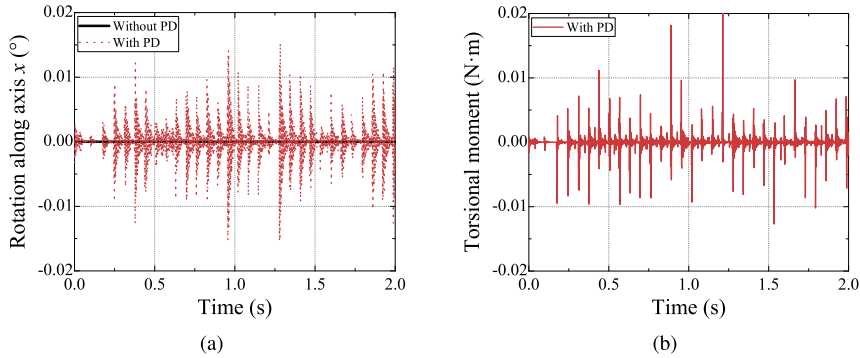
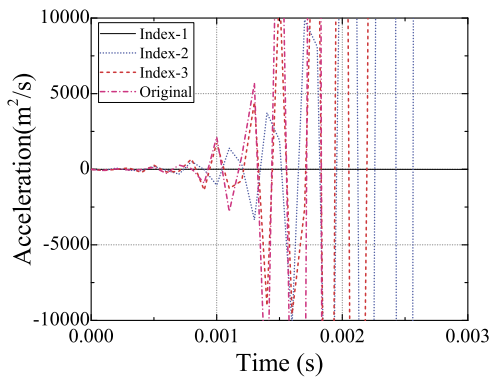


Fig. 16 Time histories of (a) rotation about axis x and (b) torsional moment in the coupled model

Fig. 17 Time histories of the accelerations using different input formats



end of the beam are given. It follows that in the coupled model, small but not negligible and physically justified out-of-plane motion and torsional moment are caused by the activity of the particles in the PD, which, of course, cannot be observed when the simplified model is used.

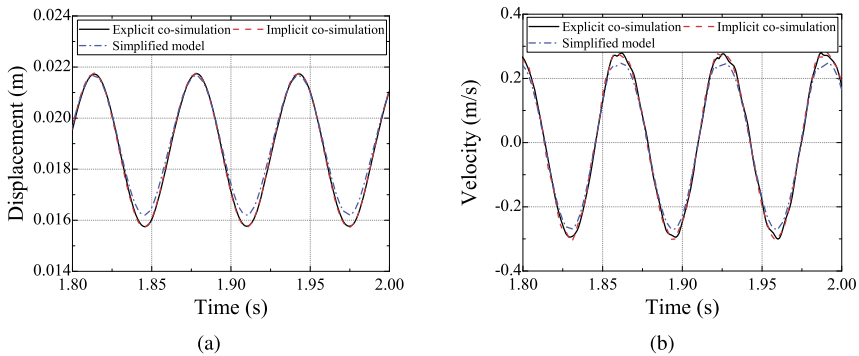


Fig. 18 Zoom pictures of time histories of (a) displacement and (b) velocity along axis y of the container using different cosimulation schemes

Table 3 Computational time of different cosimulation schemes

Cosimulation schemes	CPU time (s)	Elapsed time (s)	Chrono CPU time (s)	Iterations
Explicit cosimulation	2683.4	672.5	2571.5	20001
Implicit cosimulation	10754.9	2682.0	10693.4	60011

The explicit cosimulation schemes using the original inputs as in Eq. (20), the index-2 inputs as in Eq. (46b), and index-3 modified inputs as in Eq. (46c) are also used in the cosimulation, as shown in Fig. 17. The results show that index-1 modified inputs generate stable results, whereas the other schemes fail to obtain algorithmically stable results. Those results verify that the explicit cosimulation using index-1 inputs can improve the stability even in complex coupled systems.

The proposed explicit cosimulation method is also compared with the implicit cosimulation scheme between MBDyn and Chrono presented in [7], which adopts an iterative process to improve the algorithmic stability of the cosimulation and exchanges contact forces/moments as coupling variables. In the implicit cosimulation the tolerance $\tilde{\varepsilon}$ for the coupling variables is set as $\tilde{\varepsilon} = 0.001$, and up to 10 iterations are allowed within each step. The time step size $\Delta t = 10^{-4}$ s is used in both cosimulation schemes, and all the other conditions are the same as in the last numerical experiment. The results for the smooth/nonsmooth coupled model from different cosimulation schemes are shown in Fig. 18, and the results for the simplified model are also plotted as a reference. We can see that the developed explicit cosimulation scheme predicts results very close to the implicit one. The computational cost of different cosimulation schemes during $t \in [0, 2]$ s are listed in Table 3, where cosimulations were all run on an Intel i5-8300H CPU @ 2.30 GHz with 8G RAM and 4 threads. The CPU time denotes the amount of time that the CPU needs, whereas the elapsed time represents the actual wall clock time that the simulation requires; “Chrono CPU time” denotes the CPU time used by the nonsmooth solver; “Iterations” indicates the number of times the nonsmooth solver is called during cosimulation. It follows that under similar accuracy requirements, the proposed explicit cosimulation scheme can save almost three-quarters of the computational time required by the implicit cosimulation scheme because the nonsmooth solver is called just once for each discrete time, whereas the implicit approach requires the restart and reintegration of the nonsmooth subsystem at each step multiple times with

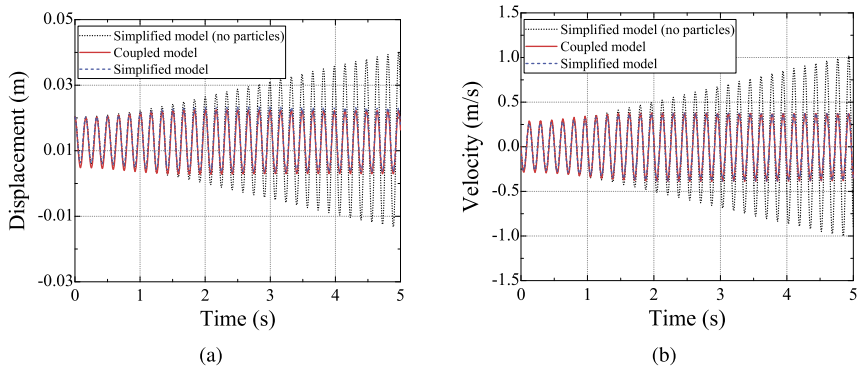


Fig. 19 Time histories of (a) displacement and (b) velocity along axis y of the container

an average of almost exactly three additional calls per time step. Therefore the explicit cosimulation scheme is recommended concerning computational cost, although the implicit one may show better stability in general applications.

The explicit cosimulation scheme using the index-1 modified inputs is also able to deal with the coupled model when the stiffness of the beam is reduced. The thickness of the beam is reduced to 1 mm with the other parameters remaining as in Table 2. The equivalent mass and stiffness of the beam are

$$M_e = 0.062621 \text{ kg}, \quad K_e = 100.8 \text{ N/m}, \quad (51)$$

which leads to a value for the first natural frequency of approximately 6.1 Hz. The dynamic response to an excitation close to the first natural frequency, obtained by setting the angular velocity of the crank to $\omega_{\text{cra}} = 6.1 \times 2\pi \text{ rad/s}$, is shown in Fig. 19. It follows that the results obtained using the cosimulation are consistent with those of the simplified model, which verifies the explicit cosimulation scheme. These results also indicate the ability of the PD to effectively produce damping and suppress vibrations in the vicinity of the first natural frequency.

5 Conclusions

An explicit cosimulation scheme is developed between an implicit solver for differential-algebraic equations, implemented in MBDyn, and a nonsmooth solver based on cone complementarity problems, implemented as a user-defined element using Project Chrono libraries, to deal with coupled smooth/nonsmooth systems with constraints. At the constraints that connect subsystems, a force-displacement decomposition technique is employed to split the whole system into smooth and nonsmooth subsystems, where smooth subsystems output kinematic variables and receive reaction forces and moments as inputs, whereas nonsmooth subsystems receive kinematic variables as inputs and send reaction forces/moments as outputs.

To improve the stability of explicit cosimulation schemes and keep the details of the nonsmooth solver confined, the inputs to the nonsmooth subsystems are modified along the lines of Baumgarte’s stabilization technique and the properties of the integrator used in the

nonsmooth solver. The modified inputs can be used to describe general coupling joints by modifying the motion of markers fixed to coupling bodies.

For the communication platform between MBDyn and Project Chrono, a loadable Chrono-Interface module has been developed in MBDyn using dynamic linking and run-time loading. The nonsmooth subsystems and interface functions are compiled as dynamical libraries that are run-time loaded into MBDyn.

Finally, a linear two-mass oscillator, a double pendulum system, and a smooth flexible system coupled with a nonsmooth particle damper are cosimulated using the explicit cosimulation scheme. For the two-mass oscillator, stability plots with the same system parameters but different input formats are also presented. It follows that using index-1 modified inputs, the largest stability region is achieved, whereas using the original inputs, algorithmically unstable results are obtained in most cases. The explicit cosimulation using various modified inputs is also compared to its implicit counterpart. The results show that explicit cosimulation requires less computational effort to obtain comparable accuracy. In the double pendulum system, cosimulation results agree well with those obtained using a monolithic simulation, which verifies that the explicit cosimulation scheme can deal with a generic coupling joint. In the system coupled with a particle damper, a simplified model of the smooth subsystem is also simulated for reference. The results of the coupled model using the explicit cosimulation scheme with index-1 modified inputs are consistent with those of the simplified model and of the coupled model using an implicit cosimulation scheme. Besides, the explicit cosimulation scheme requires much less time than the implicit one. Therefore the explicit cosimulation scheme is a better choice in terms of efficiency for solving the smooth/nonsmooth coupled systems.

Nomenclature

Formulation

α, β, γ	parameters of Baumgarte's stabilization method in the modified inputs
Δt	time step size
$(\cdot)_{/x}$	partial derivatives with respect to \mathbf{x}
$(\cdot)_n$	discretized variables at time t_n
$\Phi_{\mathbf{n}\pi_i}$	normal distance at the contact π_i
Π	set collecting all closed contacts
π_i	index referring to an element of the closed contact set Π
ρ_∞	asymptotic spectral radius of the implicit linear two-step scheme
a_i ($i = 1, 2$)	coefficients for variables \mathbf{x} in the implicit linear two-step scheme
b_j ($j = 1, 2, 3$)	coefficients for variables $\dot{\mathbf{x}}$ in the implicit linear two-step scheme
NS_j ($j = 0, 1, 2, \dots$)	bodies in the non-smooth subsystems
$p_{\mathbf{n}\pi_i}$	normal contact force along directions \mathbf{n}_{π_i} at the contact π_i
$p_{\mathbf{u}\pi_i}$	tangential contact force along directions \mathbf{u}_{π_i} at the contact π_i
$p_{\mathbf{w}\pi_i}$	tangential contact force along directions \mathbf{w}_{π_i} at the contact π_i
S_i ($i = 0, 1, 2, \dots$)	bodies in the smooth subsystems
t	time
$v_{\mathbf{n}\pi_i}$	normal velocity at the contact π_i
$v_{\mathbf{u}\pi_i}$	tangential velocity along direction \mathbf{u}_{π_i} at the contact π_i
$v_{\mathbf{w}\pi_i}$	tangential velocity along direction \mathbf{w}_{π_i} at the contact π_i
$\mathbf{a} \times$	skew-symmetric matrix of the vector $\mathbf{a} \in \mathbb{R}^3$
\mathbf{f}_1	external generalized forces vector in the smooth subsystem
\mathbf{f}_2	external generalized forces vector in the non-smooth subsystem
\mathbf{g}_1	bilateral constraints in the smooth subsystem

\mathbf{g}_2	bilateral constraints in the non-smooth subsystem
\mathbf{g}_{co}	coupling constraints between the smooth and the non-smooth subsystem
\mathbf{M}_1	mass matrix of the smooth subsystem
\mathbf{M}_2	mass matrix of the non-smooth subsystem
$\mathbf{n}_{\pi_i}, \mathbf{u}_{\pi_i}, \mathbf{w}_{\pi_i}$	three unit orthogonal vectors representing the local reference at the contact π_i
\mathbf{p}_{π_i}	the forces vector collecting the contact forces $p_{\mathbf{n}_{\pi_i}}, p_{\mathbf{u}_{\pi_i}}$, and $p_{\mathbf{w}_{\pi_i}}$ at the contact π_i
$\mathbf{q}_{1,A}^{co}$	physical coordinates of the Maker “A”
\mathbf{q}_1^{co}	physical coordinates of the coupling bodies in the smooth subsystem
$\mathbf{q}_{2,B}^{co}$	physical coordinates of the Maker “B”
\mathbf{q}_2^{co}	physical coordinates of the coupling bodies in the non-smooth subsystem
\mathbf{q}_1	physical coordinates of the smooth subsystem
\mathbf{q}_2	physical coordinates of the non-smooth subsystem
$\mathbf{r}_{1,A}^{co}, \mathbf{r}_{1,AR}^{co}$	displacement and rotation coordinates in $\mathbf{q}_{1,A}^{co}$
$\mathbf{r}_1^{co}, \mathbf{r}_{1R}^{co}$	displacement and rotation coordinates in \mathbf{q}_1^{co}
$\mathbf{r}_{2,B}^{co}, \mathbf{r}_{2,BR}^{co}$	displacement and rotation coordinates in $\mathbf{q}_{2,B}^{co}$
$\mathbf{r}_2^{co}, \mathbf{r}_{2R}^{co}$	displacement and rotation coordinates in \mathbf{q}_2^{co}
\mathbf{R}_i^{co} ($i = 1, 2$)	rotation matrices corresponding to the rotation coordinates $\mathbf{r}_{i,R}^{co}$
$\mathbf{r}_{A,loc}$	relative offsets of Marker “A” in bodies’ local reference frame
$\mathbf{r}_{B,loc}$	relative offsets of Marker “B” in bodies’ local reference frame
\mathbf{u}_1	inputs of the smooth subsystem
$\mathbf{u}_{21}, \mathbf{u}_{22}$	components of the inputs \mathbf{u}_2
\mathbf{u}_2	inputs of the non-smooth subsystem
\mathbf{v}_1	generalized velocity coordinate vector ($\mathbf{v}_1 \equiv \dot{\mathbf{q}}_1$ is assumed)
\mathbf{v}_2	generalized velocity coordinate vector ($\mathbf{v}_2 \equiv \dot{\mathbf{q}}_2$ is assumed)
$\mathbf{v}_{T\pi_i}$	tangential velocity at the contact π_i
\mathbf{x}	state variables of the smooth subsystem collecting $\mathbf{q}_1, \dot{\mathbf{q}}_1$, and λ_1
\mathbf{y}_1	outputs of the smooth subsystem
\mathbf{y}_2	outputs of the non-smooth subsystem
\mathbf{f}_2	external impulse generated by the external forces vector \mathbf{f}_2
$\tilde{\mathbf{p}}_{\pi_i}$	impulse generated from the friction contact π_i
$\mathbf{q}_1^{(0)}$	predicted physical coordinated in the smooth subsystem
$\ddot{\mathbf{q}}_1^{(0)}$	predicted generalized acceleration vector in the smooth subsystem
$\dot{\mathbf{q}}_1^{(0)}$	predicted generalized velocity vector in the smooth subsystem
λ^{co}	Lagrange multiplier of the coupling constraints
λ_1	Lagrange multiplier of the bilateral constraints \mathbf{g}_1
λ_2	Lagrange multiplier of the bilateral constraints
$\tilde{\lambda}^{co}$	reaction impulse corresponding to the coupling constraints \mathbf{g}^{co}
$\tilde{\lambda}_2$	reaction impulse corresponding to the bilateral constraints \mathbf{g}_2

Two-Mass Oscillator

λ	Lagrange multiplier of the coupling constraint in the two-mass oscillator
ω_{01})	natural frequency of the smooth subsystem
ω_{02})	natural frequency of the “non-smooth” subsystem
$\tilde{\rho}$	spectral radius of the recurrence equations of the two-mass oscillator
ξ_1	damping factor of the smooth subsystem
ξ_2	damping factor of the “non-smooth” subsystem
A_{vj}	coefficients in the characteristic equations
c_i ($i = 1, 2$)	damping coefficient of Mass i in the two-mass oscillator
k_i ($i = 1, 2$)	stiffness of Mass i in the two-mass oscillator

m_i ($i = 1, 2$)	mass of Mass i in the two-mass oscillator
u_1	inputs for the smooth subsystem in the two-mass oscillator
u_{21}, u_{22}	inputs for the “non-smooth” subsystem in the two-mass oscillator
$v_1^{(0)}$	predicted velocity of Mass 1
v_i ($i = 1, 2$)	velocity of Mass i in the two-mass oscillator
$x(t_n)$	theoretical solution at time t_n
$x_1^{(0)}$	predicted position of Mass 1
x_i ($i = 1, 2$)	position of Mass i in the two-mass oscillator
x_n	numerical solution at time t_n
$\dot{v}_1^{(0)}$	predicted position of Mass 1

Double-Pendulum System

$\alpha_{z,i}$ ($i = 1, 2$)	initial rotation about axis z in the double-pendulum system
$\omega_{z,i}$ ($i = 1, 2$)	initial angular velocity about axis z in the double-pendulum system
L	length of the pendulum in the double-pendulum system
m_i ($i = 1, 2$)	mass of Pendulum i in the double-pendulum system
$\mathbf{E}_3 \in \mathbb{R}$	identity matrix
\mathbf{J}_{C_i} ($i = 1, 2$)	moment of the inertia with respect to point C_i of Pendulum i in the double-pendulum system

Crank-Slider-PD System

$(\cdot)_1$	variables of rod 1
$(\cdot)_2$	variables of rod 2
$(\cdot)_{\text{cra}}$	variables of the crank
$(\cdot)_b$	variables of the beam
$(\cdot)_c$	variables of the container of the PD
$(\cdot)_p$	variables of the PD
$(\cdot)_s$	variables of the sensor
μ_p	friction coefficient among particles and their enclosure
ν_b	Poisson’s ratio of the beam
ω_{cra}	angular velocity of the crank
ρ	density of a body
$\tilde{\varepsilon}$	tolerance for the coupling variables in the implicit co-simulation
C_e	equivalent damping ratio of the simplified model
E_b	Young’s modulus of the beam
h_b	height of the cross section of the beam
H_c	outer height of the enclosure of the PD
$H_{c;n}$	inner height of the enclosure of the PD
I_{bz}	moment of inertia of the beam’s cross-section
J_{cx}, J_{cy}, J_{cz}	diagonal elements of \mathbf{J}_c
J_{sx}, J_{sy}, J_{sz}	diagonal elements of \mathbf{J}_s
K_e	equivalent stiffness of the simplified model
l	length of a body
m	mass of a body
M_e	equivalent mass of the simplified model
R_c	base radius of the container of the PD
R_p	radius of particles
R_r	radius of the crank, rod 1 and rod 2
w_b	width of the cross section of the beam
Y_e	excitation magnitude in the simplified model

\mathbf{J}_c the inertia tensor of the container of the PD
 \mathbf{J}_s the inertia tensor of the sensor

Acknowledgements This paper was partially funded by the European Community's Horizon 2020 Programme (H2020-EU.3.4.5.5.–ITD Engines) under Grant Agreement No. 687023 (EMS UHPE – Engine Mount System for Ultra High Pass Engine). The first author acknowledges the financial support by the China Scholarship Council and the Academic Excellence Foundation of BUAA for PhD Students, and from Google within Google Summer of Code (GSoc) 2020.

Funding Note Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References


- Masarati, P., Quaranta, G., Zanoni, A.: Dependence of helicopter pilots' biodynamic feed through on upper limbs' muscular activation patterns. *Proc. Inst. Mech. Eng., Proc., Part K, J. Multi-Body Dyn.* **227**(4), 344–362 (2013)
- Zanoni, A., Cocco, A., Masarati, P.: Multibody dynamics analysis of the human upper body for rotorcraft–pilot interaction. *Nonlinear Dyn.* **102**(3), 1517–1539 (2020)
- Bulian, G., Cercos-Pita, J.L.: Co-simulation of ship motions and sloshing in tanks. *Ocean Eng.* **152**, 353–376 (2018)
- Chung, Y.-C., Wu, Y.-R.: Dynamic modeling of a gear transmission system containing damping particles using coupled multi-body dynamics and discrete element method. *Nonlinear Dyn.* **98**(1), 129–149 (2019)
- Rakhsha, M., Pazouki, A., Serban, R., Negrut, D.: Using a half-implicit integration scheme for the SPH-based solution of fluid–solid interaction problems. *Comput. Methods Appl. Mech. Eng.* **345**, 100–122 (2019)
- Daniele, M., Quaranta, G., Masarati, P., Zanoni, A.: Pilot in the loop simulation of helicopter–ship operations using virtual reality. *Aerotec. Missili Spaz.* **99**(1), 53–62 (2020)
- Zhang, R., Zhang, H., Zanoni, A., Wang, Q., Masarati, P.: A tight coupling scheme for smooth/nonsmooth multibody co-simulation of a particle damper. *Mech. Mach. Theory* **161**, 104181 (2021)
- Rahikainen, J., González, F., Naya, M.Á.: An automated methodology to select functional co-simulation configurations. *Multibody Syst. Dyn.* **48**(1), 79–103 (2020)
- Recuero, A., Serban, R., Peterson, B., Sugiyama, H., Jayakumar, P., Negrut, D.: A high-fidelity approach for vehicle mobility simulation: Nonlinear finite element tires operating on granular material. *J. Terramech.* **72**, 39–54 (2017)
- Meyer, T., Kraft, J., Co-simulation, B.S.: Error estimation and macro-step size control. *J. Comput. Nonlinear Dyn.* **16**(4), 041002 (2021)
- Pombo, J., Ambrósio, J.: Multiple pantograph interaction with catenaries in high-speed trains. *J. Comput. Nonlinear Dyn.* **7**(4), 041008 (2012)
- Docquier, N., Lantsoght, O., Dubois, F., Brüls, O.: Modelling and simulation of coupled multibody systems and granular media using the non-smooth contact dynamics approach. *Multibody Syst. Dyn.* **49**(2), 181–202 (2020)
- Jean, M.: The non-smooth contact dynamics method. *Comput. Methods Appl. Mech. Eng.* **177**(3–4), 235–257 (1999)
- Shi, S., Gao, L., Xiao, H., Xu, Y., Yin, H.: Research on ballast breakage under tamping operation based on DEM-MBD coupling approach. *Constr. Build. Mater.* **272**, 121810 (2021)
- Xiao, W., Chen, Z., Pan, T., Li, J.: Research on the impact of surface properties of particle on damping effect in gear transmission under high speed and heavy load. *Mech. Syst. Signal Process.* **98**, 1116–1131 (2018)

16. Gagnon, L., Morandini, M., Ghiringhelli, G.L.: A review of particle damping modeling and testing. *J. Sound Vib.* **459**, 114865 (2019)
17. Peiret, A., González, F., Kövecses, J., Teichmann, M.: Co-simulation of multibody systems with contact using reduced interface models. *J. Comput. Nonlinear Dyn.* **15**(4), 041001 (2020)
18. Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. *Multibody Syst. Dyn.* **4**(2–3), 107–127 (2000)
19. Kübler, R., Schiehlen, W.: Two methods of simulator coupling. *Math. Comput. Model. Dyn. Syst.* **6**(2), 93–113 (2000)
20. Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. *Arch. Appl. Mech.* **84**(12), 1739–1769 (2014)
21. 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)
22. Schweizer, B., Lu, D.: Predictor/corrector co-simulation approaches for solver coupling with algebraic constraints. *J. Appl. Math. Mech./Z. Angew. Math. Mech.* **95**(9), 911–938 (2015)
23. Schweizer, B., Lu, D.: Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. *Multibody Syst. Dyn.* **34**(2), 129–161 (2015)
24. Meyer, T., Li, P., Lu, D., Schweizer, B.: Implicit co-simulation method for constraint coupling with improved stability behavior. *Multibody Syst. Dyn.* **44**(2), 135–161 (2018)
25. Gu, B., Asada, H.H.: Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models. *J. Dyn. Syst. Meas. Control* **126**(1), 1–13 (2004)
26. Gu, B., Gordon, B.W., Asada, H.H.: Co-simulation of coupled dynamic subsystems: a differential-algebraic approach using singularly perturbed sliding manifolds. In: *Proceedings of the 2000 American Control Conference*. ACC, IEEE Cat. No. 00CH36334, vol. 2, pp. 757–761. IEEE Press, New York (2000)
27. Schneider, F., Burger, M., Arnold, M., Simeon, B.: A new approach for force-displacement co-simulation using kinematic coupling constraints. *J. Appl. Math. Mech./Z. Angew. Math. Mech.* **97**(9), 1147–1166 (2017)
28. Negrut, D., Serban, R., Tasora, A.: Posing multibody dynamics with friction and contact as a differential complementarity problem. *J. Comput. Nonlinear Dyn.* **13**(1), 014503 (2018)
29. Tasora, A., Anitescu, M.: A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Comput. Methods Appl. Mech. Eng.* **200**(5–8), 439–453 (2011)
30. Flores, P., Leine, R., Glocker, C.: Application of the nonsmooth dynamics approach to model and analysis of the contact-impact events in cam-follower systems. *Nonlinear Dyn.* **69**(4), 2117–2133 (2012)
31. Moreau, J.J.: *Unilateral contact and dry friction in finite freedom dynamics*. In: *Nonsmooth mechanics and Applications*, pp. 1–82. Springer, Berlin (1988)
32. Tasora, A., Anitescu, M.: A convex complementarity approach for simulating large granular flows. *J. Comput. Nonlinear Dyn.* **5**(3), 031004 (2010)
33. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. *Comput. Methods Appl. Mech. Eng.* **1**(1), 1–16 (1972)
34. Masarati, P., Morandini, M., Mantegazza, P.: An efficient formulation for general-purpose multibody/multiphysics analysis. *J. Comput. Nonlinear Dyn.* **9**(4), 041001 (2014)
35. Zhang, H., Zhang, R., Zanoni, A., Masarati, P.: Performance of implicit A-stable time integration methods for multibody system dynamics. *Multibody Syst. Dyn.* **54**(3), 263–301 (2022)
36. Zhang, R., Yu, Y., Wang, Q., Wang, Q.: An improved implicit method for mechanical systems with set-valued friction. *Multibody Syst. Dyn.* **48**(2), 211–238 (2020)
37. Pfeiffer, F., Foerg, M., Ulbrich, H.: Numerical aspects of non-smooth multibody dynamics. *Comput. Methods Appl. Mech. Eng.* **195**(50–51), 6891–6908 (2006)
38. Brüls, O., Acary, V., Cardona, A.: Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized- α scheme. *Comput. Methods Appl. Mech. Eng.* **281**, 131–161 (2014)
39. Zhang, H., Zhang, R., Masarati, P.: Improved second-order unconditionally stable schemes of linear multi-step and equivalent single-step integration methods. *Comput. Mech.* **67**(1), 289–313 (2021)
40. Zhao, Z., Lu, J., Wang, Q., Liu, C., Wang, Q.: The effect of non-spherical aspect of a dimer on the dynamic behaviors. *Nonlinear Dyn.* **94**(3), 2191–2204 (2018)
41. Melanz, D., Fang, L., Jayakumar, P., Negrut, D.: A comparison of numerical methods for solving multibody dynamics problems with frictional contact modeled via differential variational inequalities. *Comput. Methods Appl. Mech. Eng.* **320**, 668–693 (2017)
42. Flores, P., Machado, M., Seabra, E., Tavares da Silva, M.: A parametric study on the Baumgarte stabilization method for forward dynamics of constrained multibody systems. *J. Comput. Nonlinear Dyn.* **6**(1), 011019 (2011)

43. Hafner, I., Popper, N.: On the terminology and structuring of co-simulation methods. In: Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, pp. 67–76 (2017)
44. Arnold, M.: Stability of sequential modular time integration methods for coupled multibody system models. *J. Comput. Nonlinear Dyn.* **5**(3), 031003 (2010)
45. Li, P., Yuan, Q.: Influence of coupling approximation on the numerical stability of explicit co-simulation. *J. Mech. Sci. Technol.* **34**, 2289–2298 (2020)
46. Schweizer, B., Li, P., Lu, D.: Explicit and implicit cosimulation methods: stability and convergence analysis for different solver coupling approaches. *J. Comput. Nonlinear Dyn.* **10**(5), 051007 (2015)
47. Luca Ghiringhelli, G., Masarati, P., Mantegazza, P.: Multibody implementation of finite volume C^0 beams. *AIAA J.* **38**(1), 131–138 (2000)
48. 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 Syst. Dyn.* **37**(1), 29–48 (2016)
49. Trigui, M., Foltete, E., Abbas, M.S., Fakhfakh, T., Bouhaddi, N., Haddar, M.: An experimental study of a multi-particle impact damper. *Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci.* **223**(9), 2029–2038 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Runsen Zhang^{1,2,3}  · Huimin Zhang⁴ · Andrea Zanoni¹ · Alessandro Tasora⁵ · Pierangelo Masarati¹

✉ R. Zhang
runsens.zhang@polimi.it

H. Zhang
huimin0327@163.com

A. Zanoni
andrea.zanoni@polimi.it

A. Tasora
alessandro.tasora@unipr.it

P. Masarati
pierangelo.masarati@polimi.it

¹ Politecnico di Milano, Milano, Italy

² CAEP Software Center for High Performance Numerical Simulation, Beijing, China

³ Institute of Applied Physics and Computational Mathematics, Beijing, China

⁴ Beijing Institute of Astronautical Systems Engineering, Beijing, China

⁵ Università di Parma, Parma, Italy