Model-based coupling for co-simulation of robotic contact tasks

Albert Peiret, Francisco González, József Kövecses,
Marek Teichmann, Andreas Enzenhoefer

Abstract

Co-simulation of complex robotic systems allows the different components to be modelled and simulated independently using methods and tools tailored to their nature and time-scale, which makes the implementation process more modular and flexible. Some applications require the use of non-iterative coupling schemes for optimal performance, such as real-time interactive environments and human and hardware-in-the-loop setups. Stability of non-iterative schemes is challenging due to the restricted and delayed information that is exchanged between subsystems, and robust prediction of interface variables is key. Here, we propose a framework for exchanging model information between mechanical systems with contact, where reduced-order models approximate the interface dynamics of the subsystems. Effective mass and force terms are formulated using a reduced representation of the model, which can then be exchanged between subsystems and integrated in their simulation. The analysis of several simulations of challenging robotic contact tasks, such as grasping and insertion with jamming, shows that model-based coupling allows for stable co-simulation with larger interface stiffness values, resulting in stronger coupling and higher simulation accuracy.

Keywords: Contact modeling, dynamics, grasping.

1 Introduction

The analysis and simulation of robotic contact tasks is an important element in design, control and training of robotic systems. The increasing complexity and level of detail demanded from the simulation of robotic mechanical systems can be addressed using co-simulation as an efficient approach for dealing with subsystems with different properties and time-scale. In co-simulation, subsystems are modelled independently and simulated using numerical methods tailored to them. Many applications of industrial interest can benefit from the use of co-simulation techniques. Notably, railway systems [3], hydraulic machinery [15], and automotive applications [5].

Robotic tasks can give rise to complex mechanical models. Decomposing such models into subsystems can have various advantages. For example, some subsystems can be modelled as articulated bodies, and simulated using efficient recursive algorithms. Contact tasks, on the other hand, require formulations that can deal with many unilateral contacts and friction efficiently. Simulating all these subsystems using a monolithic formulation may be possible in some cases, but co-simulation is often a better choice from an implementation standpoint. Specific solvers for each subsystem can be coupled together in a co-simulation setup, which can improve the implementation process, making it more modular and flexible.

Subsystem coupling is critical for a robust and stable co-simulation, and the selection of interface variables plays an important role. The best choice for the interface variables is usually not obvious, and it may depend on model characteristics as well as implementation details [15]. Subsystems exchange data at discrete communication points, and the time interval between them is known as *macro time-step*. Simulation of each subsystem is then carried out independently using different integration methods and step-sizes, also known as *micro time-steps*. Therefore, a prediction of the interface variables during the macro time-step is needed. Some schemes determine those values through iteration [24, 17], while others use extrapolation techniques. Iterative schemes are more stable than non-iterative ones [11], but at a higher computational cost that can be prohibitive for real-time applications. Furthermore, they cannot be used with subsystems that do not allow for state resetting, especially those that involve physical systems such as in human and hardware-in-the-loop simulation. This motivates the interest in developing stabilization methods for non-iterative co-simulation schemes.

Generally, coupling methods provide a prediction of the input variables of a subsystem. On the one hand, *signal-based methods* rely on past interface data and do not account for subsystem dynamics. Such methods commonly use extrapolation and filtering techniques to approximate future coupling variables. On the other hand, *model-based methods* account for the internal structure of subsystems in order to emulate subsystem dynamics and represent input/output relations (see, Fig. 1). The cornerstone of such methods is the characterization of reduced-order models, which can be obtained via model-order reduction [26]. Alternatively, system identification techniques can be used to characterize these models, which may require a learning phase to identify model parameters [21].

In our previous work [13, 14], we proposed a model-based coupling using the effective mass and effective force terms of a multibody system for the interfacing of subsystems with faster dynamics in multi-rate co-simulation setups, such as hydraulic subsystems. This also included a reduced model formulation for multibody systems with unilateral contact. In this current paper, we extend the model-based coupling concept for mechanical-mechanical co-simulation of subsystems with contact. We propose a framework where the subsystems exchange model data (i.e., effective mass and effective force terms) to formulate and integrate reduced models of other subsystems into their simulation. Simulation of complex robotic contact tasks and phenomena such as jamming illustrate the advantages of the proposed method.

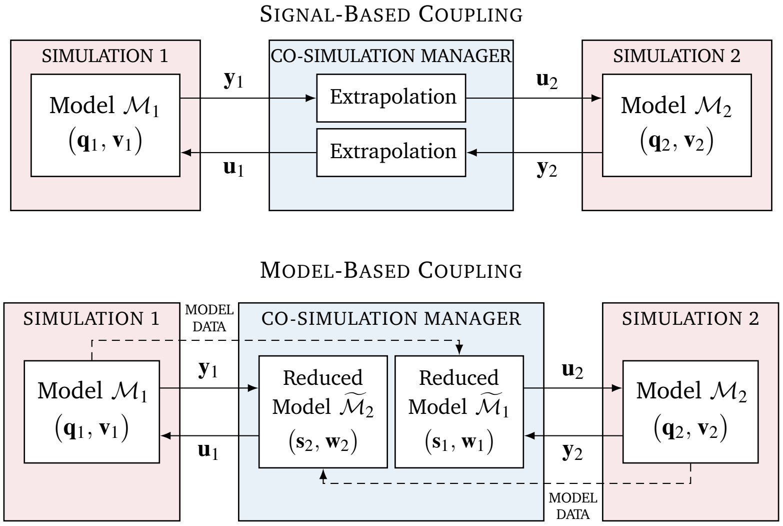

Figure 1: Co-simulation of two mechanical subsystems with signal-based coupling (top) and model-based coupling (bottom).

2 Models for systems with contact

Contact between elements in a system can be modelled with unilateral constraints, which represent contact detachment via inequalities in the constraint equations. Coulomb friction in contacts can also be modelled with constraints, where friction forces are subjected to inequalities. It is challenging to formulate contact and friction constraints in a systematic and efficient manner. In this section, we describe the main elements in complementarity formulations for systems with contact and friction.

Let us consider a constrained mechanical system parametrized by a set of p dependent generalized coordinates $\mathbf{q}$ and n dependent generalized velocities $\mathbf{v}$, which can be related as $\dot{\mathbf{q}} = \mathbf{N}\mathbf{v}$, where $\mathbf{N}(\mathbf{q})$ is the $p \times n$ transformation matrix between the two parametrizations, and $p \geq n$ in general. Constraints in the system can then be characterized by a set of r constraint velocity components

$$\eta = \mathbf{A}\mathbf{v} \tag{1}$$

where $\mathbf{A}(\mathbf{q},t)$ is the $r \times n$ constraint Jacobian matrix. In addition, holonomic constraints can also be defined at the position level by a set of r constraint coordinates $\mathbf{\Phi}(\mathbf{q},t)$ such that $\dot{\mathbf{\Phi}} = \mathbf{A}\mathbf{v} - \mathbf{b}$, where $\mathbf{b}(\mathbf{q},t) = -(\partial\mathbf{\Phi}/\partial t)$ are the prescribed constraint velocities; often $\mathbf{b} = \mathbf{0}$. Contact distances can be used as unilateral constraint coordinates, but such coordinates cannot be defined for friction constraints since tangential velocity components are generally nonholonomic.

The equations of motion of the system can be written as

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f} + \mathbf{A}^{\mathrm{T}}\boldsymbol{\lambda} \tag{2}$$

where $\mathbf{M}(\mathbf{q})$ is the $n \times n$ mass matrix, $\mathbf{f}$ is the $n \times 1$ array of generalized forces, which also include the Coriolis and centrifugal terms, and $\boldsymbol{\lambda}$ is the $r \times 1$ array of constraint interaction forces. Then, by incorporating the constraint acceleration $\dot{\eta}$, the dynamics equations of the constrained system become

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \dot{\eta} - \mathbf{g} \end{bmatrix} \tag{3}$$

where $\mathbf{g}(\mathbf{q},\mathbf{v}) = \dot{\mathbf{A}}\mathbf{v}$.

If all constraints are bilateral, then $\dot{\boldsymbol{\eta}}$ is given and the system in Eq. (3) can be solved, which has a unique solution if constraints are not redundant. On the other hand, unilateral constraints in Eq. (3) formulate a mixed *linear complementarity problem* (LCP) by introducing the complementarity condition [1]

$$\mathbf{0} \leqslant \boldsymbol{\lambda}_{\mathrm{n}} \ \perp \ \dot{\boldsymbol{\eta}}_{\mathrm{n}} \geqslant \mathbf{0} \tag{4}$$

where $\boldsymbol{\lambda}_{\mathrm{n}}$ are the normal contact force components, $\dot{\boldsymbol{\eta}}_{\mathrm{n}}$ are the normal contact acceleration components of closed contact points, and $\perp$ denotes component-wise complementarity, i.e., $\lambda_{\mathrm{n}j}\dot{w}_{\mathrm{n}j} = 0$, $\forall j$.

In general, the dynamics equations for systems with contact and Coulomb friction formulate a mixed *nonlinear complementarity problem* (NCP) [25]

$$\left.\begin{array}{r}
\begin{bmatrix} \mathbf{M} & -\mathbf{A}_{\mathrm{b}}^{\mathrm{T}} & -\mathbf{A}_{\mathrm{n}}^{\mathrm{T}} & -\mathbf{A}_{\mathrm{t}}^{\mathrm{T}} \\ \mathbf{A}_{\mathrm{b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\mathrm{n}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\mathrm{t}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \boldsymbol{\lambda}_{\mathrm{b}} \\ \boldsymbol{\lambda}_{\mathrm{n}} \\ \boldsymbol{\lambda}_{\mathrm{t}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \dot{\boldsymbol{\eta}}_{\mathrm{b}} - \mathbf{g}_{\mathrm{b}} \\ \dot{\boldsymbol{\eta}}_{\mathrm{n}} - \mathbf{g}_{\mathrm{n}} \\ \dot{\boldsymbol{\eta}}_{\mathrm{t}} - \mathbf{g}_{\mathrm{t}} \end{bmatrix} \\
\mathbf{0} \leqslant \boldsymbol{\lambda}_{\mathrm{n}} \ \perp \ \dot{\boldsymbol{\eta}}_{\mathrm{n}} \geqslant \mathbf{0} \\
\mathbf{0} \leqslant \boldsymbol{\sigma}_{\mathrm{t}}(\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{t}}) \ \perp \ \dot{\boldsymbol{\gamma}}_{\mathrm{t}} \geqslant \mathbf{0} \\
\boldsymbol{\kappa}_{\mathrm{t}}(\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{t}}, \dot{\boldsymbol{\eta}}_{\mathrm{t}}, \dot{\boldsymbol{\gamma}}_{\mathrm{t}}) = \mathbf{0}
\end{array}\right\} \tag{5}$$

where $\boldsymbol{\lambda}_{\mathrm{b}}$ are the bilateral constraint forces, $\boldsymbol{\lambda}_{\mathrm{n}}$ and $\boldsymbol{\lambda}_{\mathrm{t}}$ are the normal and tangential contact force components, respectively, and $\boldsymbol{\eta}_{\mathrm{b}} = \mathbf{A}_{\mathrm{b}}\mathbf{v}$, $\boldsymbol{\eta}_{\mathrm{n}} = \mathbf{A}_{\mathrm{n}}\mathbf{v}$, and $\boldsymbol{\eta}_{\mathrm{t}} = \mathbf{A}_{\mathrm{t}}\mathbf{v}$, are the associated constraint velocity components. Friction saturations $\boldsymbol{\sigma}_{\mathrm{t}}(\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{t}})$ define the limits of the Coulomb friction force [7]. The saturation for a contact point can be interpreted as the distance to the friction cone, which equals zero when the contact force is on the surface and positive inside the cone. For the j-th contact point, $\sigma_{\mathrm{t}_j} = \mu_j^2 \lambda_{\mathrm{n}_j}^2 - \lambda_{\mathrm{t}_{j,1}}^2 - \lambda_{\mathrm{t}_{j,2}}^2 \geqslant 0$, where μ_j is the friction coefficient. The slack velocities $\boldsymbol{\gamma}_{\mathrm{t}}$ are complementary to the saturations, and together with $\boldsymbol{\kappa}_{\mathrm{t}} = \mathbf{0}$, they ensure that the sliding velocity opposes the kinetic friction force [25].

Alternatively, the friction cone can be approximated with a pyramid to formulate a mixed LCP [7, 22, 2] Linear problems are preferable than nonlinear ones and many solver algorithms with guaranteed convergence are available in the literature [9]. Further details on the numerical methods to solve Eq. (5) are described in Section 5.

3 Co-simulation of mechanical systems

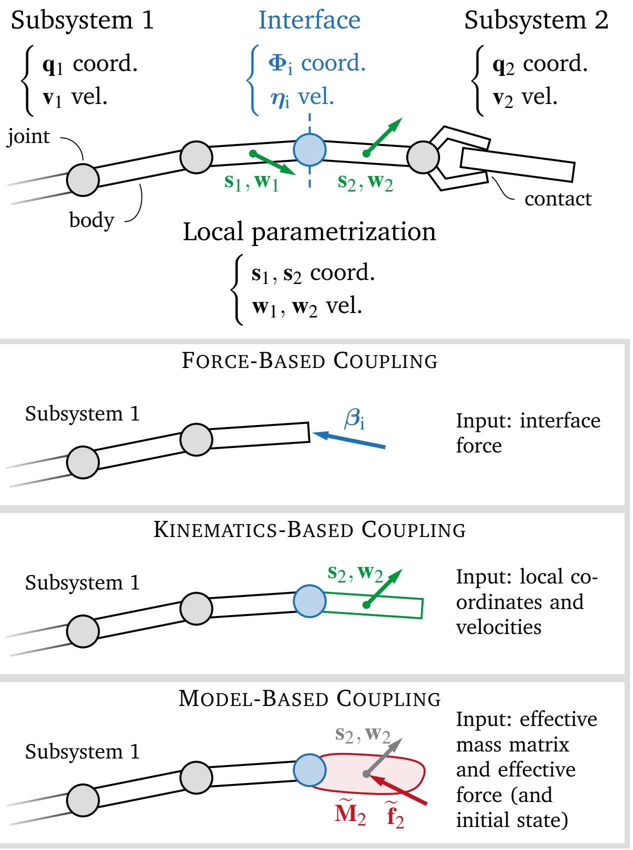
Subsystem 1
$\mathbf{q}_1$ coord.
$\mathbf{v}_1$ vel.
Interface
$\boldsymbol{\Phi}_\mathrm{i}$ coord.
η_i vel.
Subsystem 2
$\mathbf{q}_2$ coord.
$\mathbf{v}_2$ vel.
joint
body
$\mathbf{s}_1, \mathbf{w}_1$
$\mathbf{s}_2, \mathbf{w}_2$
contact
Local parametrization
$\mathbf{s}_1, \mathbf{s}_2$ coord.
$\mathbf{w}_1, \mathbf{w}_2$ vel.
FORCE-BASED COUPLING
Subsystem 1
β_i
Input: interface force
KINEMATICS-BASED COUPLING
Subsystem 1
$\mathbf{s}_2, \mathbf{w}_2$
Input: local coordinates and velocities
MODEL-BASED COUPLING
Subsystem 1
$\mathbf{s}_2, \mathbf{w}_2$
$\widetilde{\mathbf{M}}_2$
$\widetilde{\mathbf{f}}_2$
Input: effective mass matrix and effective force (and initial state)

Figure 2: Coupling methods for co-simulation of robotic systems with contact.

In this section, we describe the co-simulation of mechanical systems and the two most commonly used coupling methods: *force-based* and *kinematics-based* coupling.

Figure 2 shows a robotic system in a co-simulation setup. Let us consider two systems with n_1 and n_2 generalized velocities in $\mathbf{v}_1$ and $\mathbf{v}_2$, as well as p_1 and p_2 generalized coordinates in $\mathbf{q}_1$ and $\mathbf{q}_2$, respectively. The interface between the two systems can be characterised by a reduced parametrization with two sets of m_1 and m_2 velocities for each subsystem which can be written as

$$\mathbf{w}_1 = \mathbf{B}_1 \mathbf{v}_1 \qquad \text{and} \qquad \mathbf{w}_2 = \mathbf{B}_2 \mathbf{v}_2 \tag{6}$$

where $\mathbf{B}_1(\mathbf{q}_1)$ and $\mathbf{B}_2(\mathbf{q}_2)$ are the $m_1 \times n_1$ and $m_2 \times n_2$ Jacobian matrices of the transformation, respectively. In addition, a set of coordinates $\mathbf{s}_1(\mathbf{q}_1)$ and $\mathbf{s}_2(\mathbf{q}_2)$ can also be related to the interface velocities as $\dot{\mathbf{s}}_1 = \mathbf{R}_1\mathbf{w}_1$ and $\dot{\mathbf{s}}_2 = \mathbf{R}_2\mathbf{w}_2$. Such parametrizations generally represent the motion of the elements at the interface, such as bodies or nodes that interact with other subsystems. Moreover, the kinematics and forces of these parametrization are meant to be exchanged in a co-simulation setup as inputs and outputs, thus $m_j \ll n_j$ in general.

Interaction between the subsystems can be parametrized by a set of r_i interface velocities, which can be written in terms of the reduced parametrization as

$$\boldsymbol{\eta}_\mathrm{i} = \mathbf{G}_1\mathbf{w}_1 + \mathbf{G}_2\mathbf{w}_2 \tag{7}$$

where $\mathbf{G}_1(\mathbf{s}_1,\mathbf{s}_2)$ and $\mathbf{G}_2(\mathbf{s}_1,\mathbf{s}_2)$ are the interface Jacobian matrices. Interface velocities can also be written in terms of the global parametrization as

$$\boldsymbol{\eta}_\mathrm{i} = \mathbf{G}_1\mathbf{B}_1\mathbf{v}_1 + \mathbf{G}_2\mathbf{B}_2\mathbf{v}_2 \tag{8}$$

Additionally, a set of r_i interface coordinates $\boldsymbol{\Phi}_\mathrm{i}(\mathbf{s}_1,\mathbf{s}_2)$ can also be defined for the holonomic velocity components in $\boldsymbol{\eta}_\mathrm{i}$ such that $\dot{\boldsymbol{\Phi}}_\mathrm{i} = \boldsymbol{\eta}_\mathrm{i}$.

The set of m interface forces $\boldsymbol{\beta}_\mathrm{i}$ can then be introduced into the equations of motion of each subsystem in Eq. (2) as

$$\mathbf{M}_1\dot{\mathbf{v}}_1 = \mathbf{f}_1 + \mathbf{A}_1^\mathrm{T}\boldsymbol{\lambda}_1 + \mathbf{B}_1^\mathrm{T}\mathbf{G}_1^\mathrm{T}\boldsymbol{\beta}_\mathrm{i} \tag{9}$$

$$\mathbf{M}_2\dot{\mathbf{v}}_2 = \mathbf{f}_2 + \mathbf{A}_2^\mathrm{T}\boldsymbol{\lambda}_2 + \mathbf{B}_2^\mathrm{T}\mathbf{G}_2^\mathrm{T}\boldsymbol{\beta}_\mathrm{i} \tag{10}$$

where $\mathbf{M}_1(\mathbf{q}_1)$ and $\mathbf{M}_2(\mathbf{q}_2)$ are the mass matrices, $\mathbf{f}_1(\mathbf{q}_1,\mathbf{v}_1)$ and $\mathbf{f}_2(\mathbf{q}_2,\mathbf{v}_2)$ are the generalized forces, $\mathbf{A}_1(\mathbf{q}_1)$ and $\mathbf{A}_2(\mathbf{q}_2)$ are the constraint Jacobian matrices, and $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are the internal constraint forces of each subsystem respectively, which can also represent contact and friction as discussed in Section 2. The dynamics of each subsystem formulate a complementarity problem if contact and friction are present (see Eq. (5)), which cannot be solved without an estimation of the interface variables (e.g., interface force $\boldsymbol{\beta}_\mathrm{i}$ or interface kinematics $\boldsymbol{\eta}_\mathrm{i}$ and $\boldsymbol{\Phi}_\mathrm{i}$).

Constraints do not require any knowledge about the constitutive properties of the system and are commonly used to model interactions between elements of a system. In co-simulation, however, constraints cannot be imposed between subsystems in a systematic manner, mainly because the simulations run independently and only exchange restricted information. Formulating the subsystem dynamics with interface constraints such as

$$\boldsymbol{\eta}_{\mathrm{i}} = \mathbf{0} \tag{11}$$

results in algebraic loops in the system [11]. Thus, constraint equations require iterative co-simulation schemes to solve for algebraic loops, such as predictor–corrector schemes [18, 20].

Non-iterative co-simulation schemes for mechanical systems often exchange interface forces in order to remove algebraic loops from the system. Then, interface forces can be determined in one of the subsystems and then applied to the other subsystem as an input. Forces can be determined in one subsystem through the kinematic constraints imposed by the motion of the other subsystem. However, deciding which subsystem imposes its motion to another can largely affect the stability of these kinds of schemes [4], which often require constraint relaxation [17, 19, 6]. Therefore, interface forces can be defined through a constitutive relation using the interface kinematics as

$$\boldsymbol{\beta}_{\mathrm{i}} = -\mathbf{K}_{\mathrm{i}}\boldsymbol{\Phi}_{\mathrm{i}} - \mathbf{D}_{\mathrm{i}}\boldsymbol{\eta}_{\mathrm{i}} \tag{12}$$

where $\mathbf{K}_{\mathrm{i}}$ and $\mathbf{D}_{\mathrm{i}}$ are the interface stiffness and damping matrices, respectively. This expression can be used with any of the subsystems to determine the interface force, but the choice is difficult to generalize. Alternatively, a preliminary interface solve at the beginning of each macro time-step can determine the interface force to be applied to both subsystems, but that generally reduces the stability [16].

3.1 Force-based coupling

If the interface force $\boldsymbol{\beta}_{\mathrm{i}}^{\mathrm{In}}$ is given as an input to subsystem 1, it can be introduced into the equations of motion as an applied force

$$\mathbf{M}_1\dot{\mathbf{v}}_1 = \mathbf{f}_1 + \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\lambda}_1 + \mathbf{B}_1^{\mathrm{T}}\mathbf{G}_1^{\mathrm{T}}\boldsymbol{\beta}_{\mathrm{i}}^{\mathrm{In}} \tag{13}$$

The local position $\mathbf{s}_2^{\mathrm{In}}$ is also given to compute the Jacobian matrices $\mathbf{G}_1(\mathbf{s}_1, \mathbf{s}_2^{\mathrm{In}})$, which ultimately define the geometry of the interface force. Then, by adding the internal constraints, the dynamics formulation in Eq. (3) becomes

$$\begin{bmatrix} \mathbf{M}_1 & -\mathbf{A}_1^{\mathrm{T}} \\ \mathbf{A}_1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_1 \\ \boldsymbol{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 + \mathbf{B}_1^{\mathrm{T}}\mathbf{G}_1^{\mathrm{T}}\boldsymbol{\beta}_{\mathrm{i}}^{\mathrm{In}} \\ \dot{\boldsymbol{\eta}}_1 - \mathbf{g}_1 \end{bmatrix} \tag{14}$$

Instability issues in this formulation come from the fact that no specification is made on the interface kinematics, which are often used as output variables in some formulations, i.e., $\mathbf{w}_1^{\mathrm{Out}}$ and $\mathbf{s}_1^{\mathrm{Out}}$ are not constrained. Therefore, the motion of the elements at the interface is independent from that of the other subsystem, leading to significant error in the interface kinematics $\boldsymbol{\eta}_{\mathrm{i}}$ and $\boldsymbol{\Phi}_{\mathrm{i}}$, as well as undesired high-frequency oscillations.

3.2 Kinematics-based coupling

If the kinematics of subsystem 2 is given as input to subsystem 1 (i.e., $\mathbf{s}_2^{\mathrm{In}}$, $\mathbf{w}_2^{\mathrm{In}}$ and $\dot{\mathbf{w}}_2^{\mathrm{In}}$), constraints can be used to define the interface kinematics. Differentiation of the interface velocity in Eq. (7) yields the interface acceleration

$$\dot{\boldsymbol{\eta}}_{\mathrm{i}} = \mathbf{G}_1\dot{\mathbf{w}}_1 + \mathbf{G}_2\dot{\mathbf{w}}_2^{\mathrm{In}} + \mathbf{g}_{\mathrm{i}}^{\mathrm{In}} \tag{15}$$

where $\mathbf{g}_{\mathrm{i}}^{\mathrm{In}}(\mathbf{s}_1, \mathbf{w}_1, \mathbf{s}_2^{\mathrm{In}}, \mathbf{w}_2^{\mathrm{In}}) = \dot{\mathbf{G}}_1\mathbf{w}_1 + \dot{\mathbf{G}}_2\mathbf{w}_2$ are nonlinear terms that also depend on the input kinematics.

The dynamics equations of the constrained subsystem 1 can be written as

$$\begin{bmatrix} \mathbf{M}_1 & -\mathbf{A}_1^{\mathrm{T}} & -\mathbf{B}_1^{\mathrm{T}}\mathbf{G}_1^{\mathrm{T}} \\ \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_1\mathbf{B}_1 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_1 \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\beta}_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \dot{\boldsymbol{\eta}}_1 - \mathbf{g}_1 \\ \dot{\boldsymbol{\eta}}_{\mathrm{i}} - \mathbf{g}_{\mathrm{i}}^{\mathrm{In}} - \mathbf{G}_2\dot{\mathbf{w}}_2^{\mathrm{In}} \end{bmatrix} \tag{16}$$

where the motion of the interface $\boldsymbol{\eta}_{\mathrm{i}}$ must be specified either by equalities or inequalities, in case of contact. Furthermore, the input local configuration $\mathbf{s}_2^{\mathrm{In}}$ and velocity $\mathbf{w}_2^{\mathrm{In}}$ can be extrapolated using the input acceleration $\dot{\mathbf{w}}_2^{\mathrm{In}}$ along with the time integration of subsystem 1 during the macro time-step. As a result, simulation of subsystem 1 determines the interface constraint force $\boldsymbol{\beta}_{\mathrm{i}}^{\mathrm{Out}}$, which can then be passed to subsystem 2.

Unfortunately, interface constraints are usually not stable and they require some relaxation.

Thus, the interface force β_i can be written in terms of the interface kinematics using the expression in Eq. (12), so that the equations of motion can be written as

$$\mathbf{M}_1\dot{\mathbf{v}}_1 = \mathbf{f}_1 + \mathbf{A}_1^\mathrm{T}\boldsymbol{\lambda}_1 + \mathbf{B}_1^\mathrm{T}\mathbf{G}_1^\mathrm{T}\beta_\mathrm{i}(\mathbf{s}_1, \mathbf{w}_1, \mathbf{s}_2^\mathrm{In}, \mathbf{w}_2^\mathrm{In}) \tag{17}$$

which can be solved implicitly using constraint regularization, for instance.

4 Model-based coupling

The main idea behind model-based coupling is to obtain a reduced-order model of a subsystem that can emulate its dynamics at the co-simulation interface. Reduced models can provide an approximation of the subsystem outputs in terms of the inputs, which can be used to extrapolate the coupling variables [21] or even replace the subsystem during a macro time-step [13]. Here, we formulate the dynamics equations of the reduced model using the local parametrization $\mathbf{w}$ in Eq. (6). Contact and friction within the subsystem are taken into account by satisfying constraint complementarity, as in [14]. Then, we exchange the dynamic equations of the reduced model with other subsystems and integrate them in their simulation.

Let us then select local parametrization $\mathbf{w}_2 = \mathbf{B}_2\mathbf{v}_2$ to formulate the dynamics of subsystem 2 as [13]

$$\widetilde{\mathbf{M}}_2\dot{\mathbf{w}}_2 = \widetilde{\mathbf{f}}_2 + \mathbf{G}_2^\mathrm{T}\beta_\mathrm{i} \tag{18}$$

where $\widetilde{\mathbf{M}}_2(\mathbf{q}_2)$ is the $m_2 \times m_2$ effective mass matrix of subsystem 2 in the local parametrization $\mathbf{w}_2$, and $\widetilde{\mathbf{f}}_2(\mathbf{q}_2, \mathbf{v}_2)$ is the $m_2 \times 1$ effective force.

The effective mass depends on the configuration of the subsystem as well as the interactions of the elements within that subsystem. By decoupling the interface motion from the subsystem motion compatible with the internal constraints, an expression for the effective mass can be obtained as [13]

$$\widetilde{\mathbf{M}}_2 = \left(\mathbf{B}_2(\mathbf{I} - \mathbf{P}_2)\mathbf{M}_2^{-1}\mathbf{B}_2^\mathrm{T}\right)^{-1} \tag{19}$$

where $\mathbf{P}_2(\mathbf{q}_2)$ is an $n_2 \times n_2$ matrix that accounts for the internal constraints and the constrained motion, and $\mathbf{I}$ is the identity matrix. In the absence of internal constraints, $\mathbf{P}_2 = \mathbf{0}$ and the effective mass $\widetilde{\mathbf{M}}_2 = \left(\mathbf{B}_2\mathbf{M}_2^{-1}\mathbf{B}_2^{\mathrm{T}}\right)^{-1}$ [10]. On the other hand, if all internal constraints are bilateral ($\dot{\eta} = \mathbf{0}$) the projector matrix can be written as $\mathbf{P}_2 = \mathbf{M}_2^{-1}\mathbf{A}_2^{\mathrm{T}}\left(\mathbf{A}_2\mathbf{M}_2^{-1}\mathbf{A}_2^{\mathrm{T}}\right)^{-1}\mathbf{A}_2$ [13].

However, unilateral contact and friction may violate the constraint equation $\dot{\eta} = \mathbf{0}$ due to complementarity in Eq. (5) if the force limit is reached, e.g., a contact can detach if the force is zero. Therefore, constraints with force limits must be treated differently [12]. Closed contacts ($\dot{\eta}_{\mathrm{n}} = \mathbf{0}$) and static friction ($\eta_{\mathrm{t}}, \dot{\eta}_{\mathrm{t}} = \mathbf{0}$) can be considered as constraint forces, we call them *active constraints* [12]. Open contacts and kinetic friction are not constraints any more. We remove open contact from the effective mass formulation and consider kinetic friction as an applied force.

Hence, the projector operator [14]

$$\mathbf{P}_2 = \mathbf{M}_2^{-1}\hat{\mathbf{A}}_2^{\mathrm{T}}\left(\hat{\mathbf{A}}_2\mathbf{M}_2^{-1}\hat{\mathbf{A}}_2^{\mathrm{T}}\right)^{-1}\hat{\mathbf{A}}_2 \tag{20}$$

where $\hat{\mathbf{A}}_2$ is the Jacobian matrix of the active constraints (i.e., bilateral, closed contacts, and static friction). Likewise, the effective force can be written as [14]

$$\widetilde{\mathbf{f}}_2 = \widetilde{\mathbf{M}}_2\Big(\mathbf{B}_2(\mathbf{I}-\mathbf{P}_2)\mathbf{M}_2^{-1}(\mathbf{f}_2 + \bar{\mathbf{A}}_2^{\mathrm{T}}\bar{\boldsymbol{\lambda}}_2) + \dot{\mathbf{B}}_2\mathbf{v}_2 + \mathbf{B}_2\mathbf{P}_2\dot{\mathbf{v}}_2\Big) \tag{21}$$

where $\bar{\mathbf{A}}_2$ is the Jacobian matrix of the constraints removed from the active set (i.e., sliding contacts) and $\bar{\boldsymbol{\lambda}}_2$ are the kinetic friction force components.

The reduced model of subsystem 2 in Eq. (18) may then be combined with subsystem 1, such that the dynamics equations of the combined system can be written as

$$\begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & -\mathbf{A}_1^{\mathrm{T}} & -\mathbf{B}_1^{\mathrm{T}}\mathbf{G}_1^{\mathrm{T}} \\ \mathbf{0} & \widetilde{\mathbf{M}}_2^{\mathrm{In}} & \mathbf{0} & -\mathbf{G}_2^{\mathrm{T}} \\ \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_1\mathbf{B}_1 & \mathbf{G}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_1 \\ \dot{\mathbf{w}}_2 \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\beta}_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \widetilde{\mathbf{f}}_2^{\mathrm{In}} \\ \dot{\eta}_1 - \mathbf{g}_1 \\ \dot{\eta}_{\mathrm{i}} - \mathbf{g}_{\mathrm{i}} \end{bmatrix} \tag{22}$$

where $\widetilde{\mathbf{M}}_2^{\mathrm{In}}$ and $\widetilde{\mathbf{f}}_2^{\mathrm{In}}$ are the input of subsystem 1 and updated every macro time-step. In addition, the reduced parametrization $\mathbf{s}_2$ and $\mathbf{w}_2$ is updated every micro time-step, and it evolves according to the dynamics of both subsystem.

The main advantage of this formulation is that the interface constraints are solved inside the subsystem. This allows us to use constraint regularization for the interface constraints and ultimately improve the stability of mechanical–mechanical subsystem coupling in co-simulation.

5 Numerical formulation

The dynamics of systems with contacts are often formulated at the impulse–momentum level for a more consistent treatment of the nonsmooth dynamic equations [1]. The system dynamics equations are then discretized in time and the generalized acceleration can be approximated by a finite difference of the generalized velocity as

$$\dot{\mathbf{v}} = \frac{\mathbf{v}^{k+1} - \mathbf{v}^{k}}{h} \tag{23}$$

where $\mathbf{v}^k$ is the known velocity at the beginning of the time-step, $\mathbf{v}^{k+1}$ is the unknown velocity at the end of the times-step, and h is the simulation micro step-size. Thus, the dynamics equations in Eq. (3) can be written as [14]

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{k+1} \\ \hat{\boldsymbol{\lambda}}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{k} \\ \boldsymbol{\eta}^{k+1} \end{bmatrix} \tag{24}$$

where $\mathbf{p}^k(\mathbf{q}^k, \mathbf{v}^k) = \mathbf{M}(\mathbf{q}^k)\mathbf{v}^k + h\mathbf{f}(\mathbf{q}^k, \mathbf{v}^k)$ is the known generalized momentum, $\mathbf{M}$ and $\mathbf{A}$ are the known mass and Jacobian matrices evaluated with the known coordinates $\mathbf{q}^k$, and $\hat{\boldsymbol{\lambda}}^{k+1}$ is the unknown constraint impulse over the micro time-step, which can be related to the unknown constraint force as $h\boldsymbol{\lambda}^{k+1}$. Here, the constraint velocity $\boldsymbol{\eta} = \mathbf{A}\mathbf{v}$ represents the internal subsystem constraints $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$, as well as the interface constraints $\boldsymbol{\eta}_{\mathrm{i}}$.

As discussed in Section 2, contact and friction can be described by limits in the constraint forces $\boldsymbol{\lambda}$. Linear force limits can be generally written as

$$\boldsymbol{\lambda}^{\mathrm{low}} \leqslant \boldsymbol{\lambda}^{k+1} \leqslant \boldsymbol{\lambda}^{\mathrm{upp}} \tag{25}$$

where $\boldsymbol{\lambda}^{\mathrm{low}}$ and $\boldsymbol{\lambda}^{\mathrm{upp}}$ are the lower and upper limits of the constraint forces, respectively. Unilateral constraints describing normal contact forces $\boldsymbol{\lambda}_{\mathrm{n}}$ are characterized by the limits $\boldsymbol{\lambda}_{\mathrm{n}}^{\mathrm{low}} = \mathbf{0}$ and $\boldsymbol{\lambda}_{\mathrm{n}}^{\mathrm{upp}} = +\infty$, whereas bilateral constraint force limits can be set to infinity, i.e., $\boldsymbol{\lambda}_{\mathrm{b}}^{\mathrm{low}} = -\infty$ and

$\boldsymbol{\lambda}_{\mathrm{b}}^{\mathrm{upp}} = +\infty$. Limits in Coulomb friction forces, on the other hand, need to be linearised in order to formulate an LCP, which can be written component-wise as $\boldsymbol{\lambda}_{\mathrm{t}}^{\mathrm{low}} = -\boldsymbol{\mu}\boldsymbol{\lambda}_{\mathrm{n}}^{k}$ and $\boldsymbol{\lambda}_{\mathrm{t}}^{\mathrm{upp}} = +\boldsymbol{\mu}\boldsymbol{\lambda}_{\mathrm{n}}^{k}$, where $\boldsymbol{\mu}$ is a matrix with friction coefficients, and $\boldsymbol{\lambda}_{\mathrm{n}}^{k}$ is the known normal contact force from the previous time-step. This description of the friction force bounds represents a linear approximation of the friction cone by a polyhedra, also known as box friction model [14].

The force limits in Eq. (25) can be enforced by two sets of non-negative saturation impulses

$$\left.\begin{aligned} \hat{\boldsymbol{\sigma}}^{\mathrm{low}} &= \hat{\boldsymbol{\lambda}}^{k+1} - \hat{\boldsymbol{\lambda}}^{\mathrm{low}} \geqslant \mathbf{0} \\ \hat{\boldsymbol{\sigma}}^{\mathrm{upp}} &= \hat{\boldsymbol{\lambda}}^{\mathrm{upp}} - \hat{\boldsymbol{\lambda}}^{k+1} \geqslant \mathbf{0} \end{aligned}\right\} \tag{26}$$

which are complementary to the slack velocities $\boldsymbol{\gamma}^{\mathrm{low}} \geqslant \mathbf{0}$ and $\boldsymbol{\gamma}^{\mathrm{upp}} \geqslant \mathbf{0}$, respectively. Slack velocities define the constraint velocity as

$$\boldsymbol{\eta}^{k+1} - \mathbf{b}^{k} = \boldsymbol{\gamma}^{\mathrm{low}} - \boldsymbol{\gamma}^{\mathrm{upp}} \tag{27}$$

where $\mathbf{b}^{k}(\mathbf{q}^{k}, t^{k})$ is the prescribed constraint velocity. Contact and friction constraints require the slack velocities to enforce the force limits defined by the saturations, but bilateral constraint equations can be simply written as $\boldsymbol{\eta}_{\mathrm{b}}^{k+1} = \mathbf{b}_{\mathrm{b}}^{k}$.

The unknown velocities $\mathbf{v}^{k+1}$ can be eliminated from Eq. (24), and by introducing the slack velocities in in Eq. (27) and the complementarity conditions between slack velocities and slack forces in Eq. (26), a mixed (or bounded) LCP can be formulated as

$$\left.\begin{aligned} &\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^{\mathrm{T}}\right)\hat{\boldsymbol{\lambda}}^{k+1} + \mathbf{z} = \boldsymbol{\gamma}^{\mathrm{low}} - \boldsymbol{\gamma}^{\mathrm{upp}} \\ &\mathbf{0} \leqslant \boldsymbol{\gamma}^{\mathrm{low}} \perp \left(\hat{\boldsymbol{\lambda}}^{k+1} - \hat{\boldsymbol{\lambda}}^{\mathrm{low}}\right) \geqslant \mathbf{0} \\ &\mathbf{0} \leqslant \boldsymbol{\gamma}^{\mathrm{upp}} \perp \left(\hat{\boldsymbol{\lambda}}^{\mathrm{upp}} - \hat{\boldsymbol{\lambda}}^{k+1}\right) \geqslant \mathbf{0} \end{aligned}\right\} \tag{28}$$

where $\mathbf{z} = \mathbf{A}\mathbf{M}^{-1}\mathbf{p}^{k} - \mathbf{b}^{k}$ is known.

Once the constraint impulses $\hat{\boldsymbol{\lambda}}^{k+1}$ are determined, the velocity at the end of the step can be obtained from Eq. (24) as

$$\mathbf{v}^{k+1} = \mathbf{M}^{-1}\left(\mathbf{p}^{k} + \mathbf{A}^{\mathrm{T}}\hat{\boldsymbol{\lambda}}^{k+1}\right) \tag{29}$$

and the position from the relation $\dot{\mathbf{q}} = \mathbf{N}\mathbf{v}$ as

$$\mathbf{q}^{k+1} = \mathbf{q}^{k} + h\mathbf{N}(\mathbf{q}^{k})\mathbf{v}^{k+1} \tag{30}$$

which is a *semi-implicit* integration scheme.

A unique solution to the mixed LCP in Eq. (28) exists if the matrix $\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^{\mathrm{T}}$ is a P-matrix (i.e., all principal minors are positive) [8], which is the case for symmetric positive-definite matrices. However, constraint redundancy produces low-rank Jacobian matrices, and consequently, positive semi-definite LCP matrices. Thus, constraint forces are not unique if constraints are redundant. Moreover, solver algorithms converge poorly if the matrix is ill-conditioned or close to singular.

6 Constraint regularization

Regularization (or relaxation) can improve numerical stability and convergence of solver algorithms [23] as well as co-simulation schemes [19]. We define constraint forces through a constitutive relation

$$\boldsymbol{\lambda}^{k+1} = -\mathbf{K}\boldsymbol{\Phi}^{k+1} - \mathbf{D}\dot{\boldsymbol{\Phi}}^{k+1} \tag{31}$$

The constraint coordinate and velocity can be approximated by the following finite differences $\boldsymbol{\Phi}^{k+1} = \boldsymbol{\Phi}^{k} + h\dot{\boldsymbol{\Phi}}^{k+1}$, and a new expression for the constraint equation can be obtained as

$$\dot{\boldsymbol{\Phi}}^{k+1} + \mathbf{C}\hat{\boldsymbol{\lambda}}^{k+1} + \boldsymbol{\varphi}^{k} = \mathbf{0} \tag{32}$$

where $\mathbf{C} = (h^2\mathbf{K} + h\mathbf{D})^{-1}$ and $\boldsymbol{\varphi}^{k} = (h\mathbf{I} + \mathbf{K}^{-1}\mathbf{D})^{-1}\boldsymbol{\Phi}^{k}$ contain the so-called regularization terms. Stiffness and damping matrices are usually diagonal, since they are usually defined by the analyst.

As in Eq. (27), slack velocities can be used to define the constraint violation $\dot{\boldsymbol{\Phi}}^{k+1} = \boldsymbol{\eta}^{k+1} - \mathbf{b}^{k} - (\boldsymbol{\gamma}^{\mathrm{low}} - \boldsymbol{\gamma}^{\mathrm{upp}})$ in Eq. (32). Then, the new formulation with the regularized constraint equations can be written as

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{k+1} \\ \hat{\boldsymbol{\lambda}}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{k} \\ \mathbf{b}^{k} - \varphi^{k} + \gamma^{\mathrm{low}} - \gamma^{\mathrm{upp}} \end{bmatrix} \tag{33}$$

which can be reduced, as in Eq. (28), to

$$\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}\right)\hat{\boldsymbol{\lambda}}^{k+1} + \mathbf{z} = \gamma^{\mathrm{low}} - \gamma^{\mathrm{upp}} \tag{34}$$

where $\mathbf{z} = \mathbf{A}\mathbf{M}^{-1}\mathbf{p}^{k} - \mathbf{b}^{k} + \varphi^{k}$. The coefficient matrix $\mathbf{A}\mathbf{M}^{-1}\mathbf{A}^{\mathrm{T}} + \mathbf{C}$ will never become rank deficient due to the term $\mathbf{C}$, which can also improve the matrix condition number.

7 Examples

Several numerical experiments in co-simulation setups were carried out with a model of a 7-DOF robotic arm performing two contact tasks in microgravity: payload grasping task, and insertion task with jamming. We compare different combinations of the coupling methods presented in this paper: *kinematics–force*, *model–force*, and *kinematics–model*, and *model–model*. The first coupling method corresponds to subsystem 1 (i.e., end effector and environment), and the second one to subsystem 2 (i.e., articulated arm).

The interface is the 6 DOF joint between the end effector and the arm, and contact only occurs between the end effector and the environment. The motion of the bodies immediately after the interface define the local parametrizations with 6 velocities and 12 coordinates each (3 translation vector components and 9 rotation matrix coefficients). Therefore, the reduced models generated have a 6×6 effective mass matrix $\widetilde{\mathbf{M}}$, which are added into the simulation as additional rigid bodies with a general mass matrix. Moreover, each link of the arm is modelled as a 6 DOF rigid body (42 DOF for the arm) and the joints between them remove 5 relative DOF using constraints, which results in a significant reduction of degrees of freedom in the reduced model.

Both micro and macro step-sizes are equal to 1 ms, and the reference solution is a monolithic simulation of the entire system with the same step-size. The co-simulation setup was implemented in C++ using Vortex simulation software as the multibody dynamics engine and contact detection algorithm, and the Eigen library for linear algebra to preform efficient sparse matrix and vector operations.

7.1 Grasping task

A gripper attached to the end effector is used to grasp a payload and displaces it with a translation (see Fig. 3). During the first second of the simulation, the gripper closes its claws and grasps the fixture attached to the payload. Then, the joints are actuated to move the load to a different location, where it is kept stable until the end of the simulation.

Figure 4 shows the position and velocity of the griper in the direction of the translation. For small values of the interface stiffness, all coupling methods are in agreement. However, for large stiffness values, simulation stability is lost for all the methods except for model–model. Kinematics-based coupling can keep the position stable in spite of high-frequency oscillations in the velocity, while the force-based coupling fails.

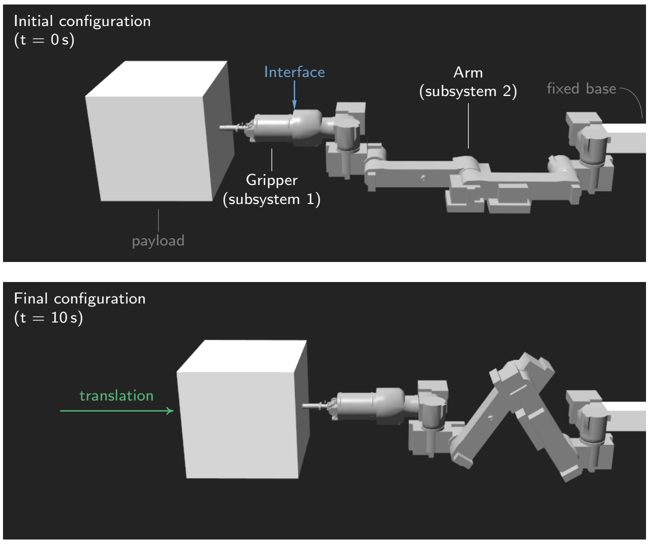

Figure 3: Model of a robotic arm grasping a payload. The joint actuation law follows a sine-square function with a peak velocity of 0.5 rad/s for the first joint.

7.2 Insertion task

The robotic arm performs the insertion of a connector into a fixed socket (see Fig. 5). The connector is rigidly attached to the end effector and has two parallel square-section pins. The

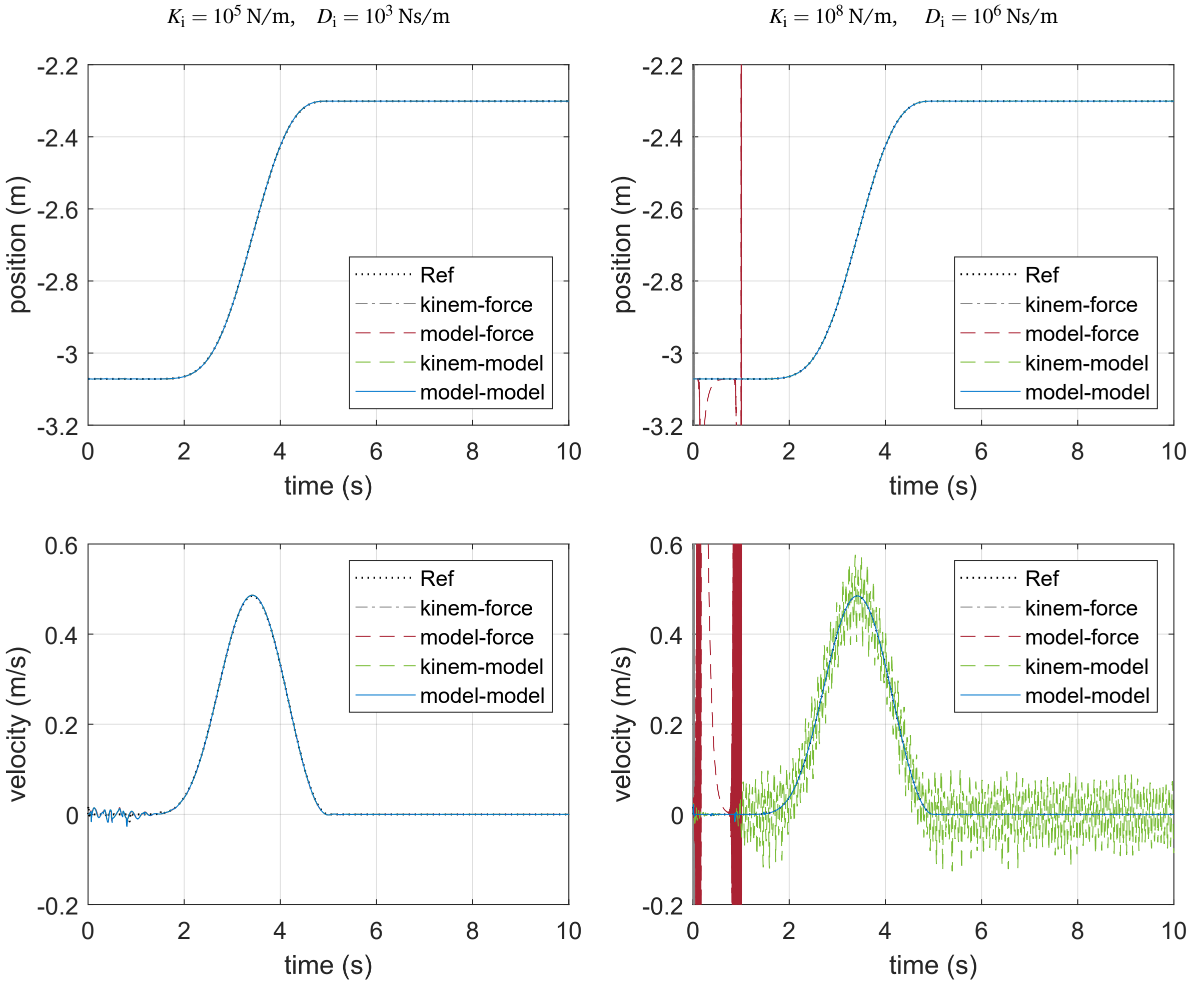

Figure 4: Position and velocity of the end effector during the grasping tasks for two sets of interface stiffness K_i and damping D_i values.

width of each pin is 100 mm, the play between pin and hole is 4 mm, and the friction coefficient is $\mu = 0.5$. Two phases precede the insertion: a 90° rotation to orient the connector followed by a translation to align the connector with the socket. Then, the connector is pushed inside the socket at a constant speed of 0.1 m/s until it stops. Jamming during the insertion phase was achieved by a small misalignment of the connector control velocity of $\alpha = 2.5°$, which caused the motion of the connector to stop before the end of the insertion phase.

Figure 6 shows the vertical position and velocity of the connector during the insertion task. For small interface stiffness values, all the methods produce similar results, but they disagree with the baseline in that jamming does not occur. On the other hand, larger stiffness values allows the model–model simulation to jam, while the other methods fail. Interestingly, capturing complex contact phenomena such as jamming requires a strong coupling between subsystems, which was attained with model-based coupling and a large interface stiffness.

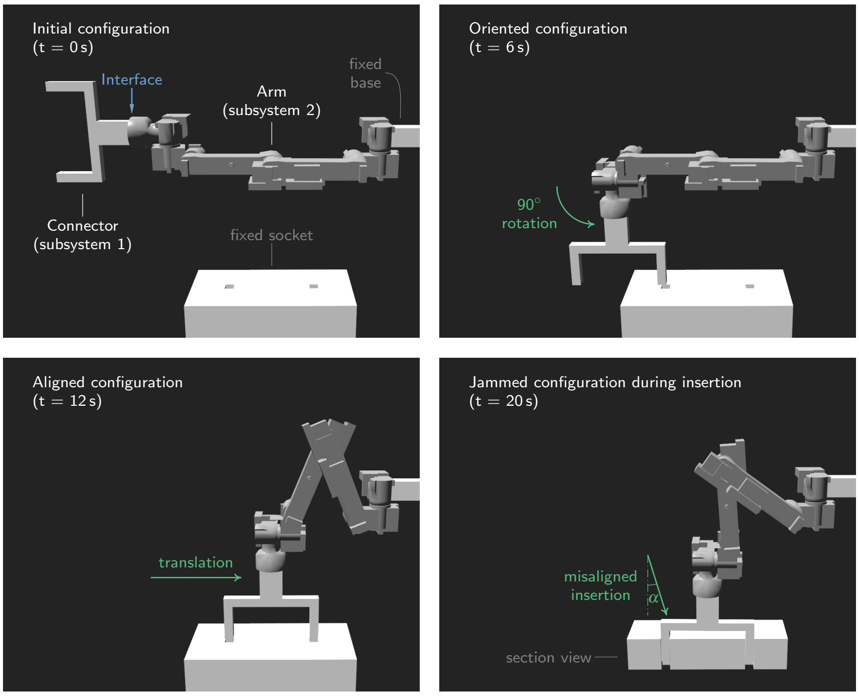

Figure 5: Model of a robotic arm performing an insertion task with jamming of the connector in the socket.

8 Discussion

Model-based coupling can substantially improve simulation accuracy compared to other signal-based coupling approaches. Large interface stiffness values reduced the error several orders of magnitude while keeping the model–model simulation method stable (see Fig. 7). The purely signal-based kinematics–force method shows the lowest stability threshold value for the interface stiffness. In some cases, introducing model-based coupling to only one of the subsystems can improve simulation stability and accuracy. But the better choice between force-based and kinematics-based coupling seems to be problem dependent.

The computational time needed to generate the reduced model is a considerable fraction of the total simulation time (Fig. 8). In the examples presented here, the reduced model generation time is less than the dynamics solver time, and the total time is still below the step size ($1\,ms$), which makes it suitable for real-time applications. Thus, model-based coupling can significantly improve simulation accuracy at a fair computational cost. To further optimize the computation, the reduced model could be generated in parallel with some other tasks, such as with the collision detection.

Figure 9 shows a correlation between the model generation time and the number of active

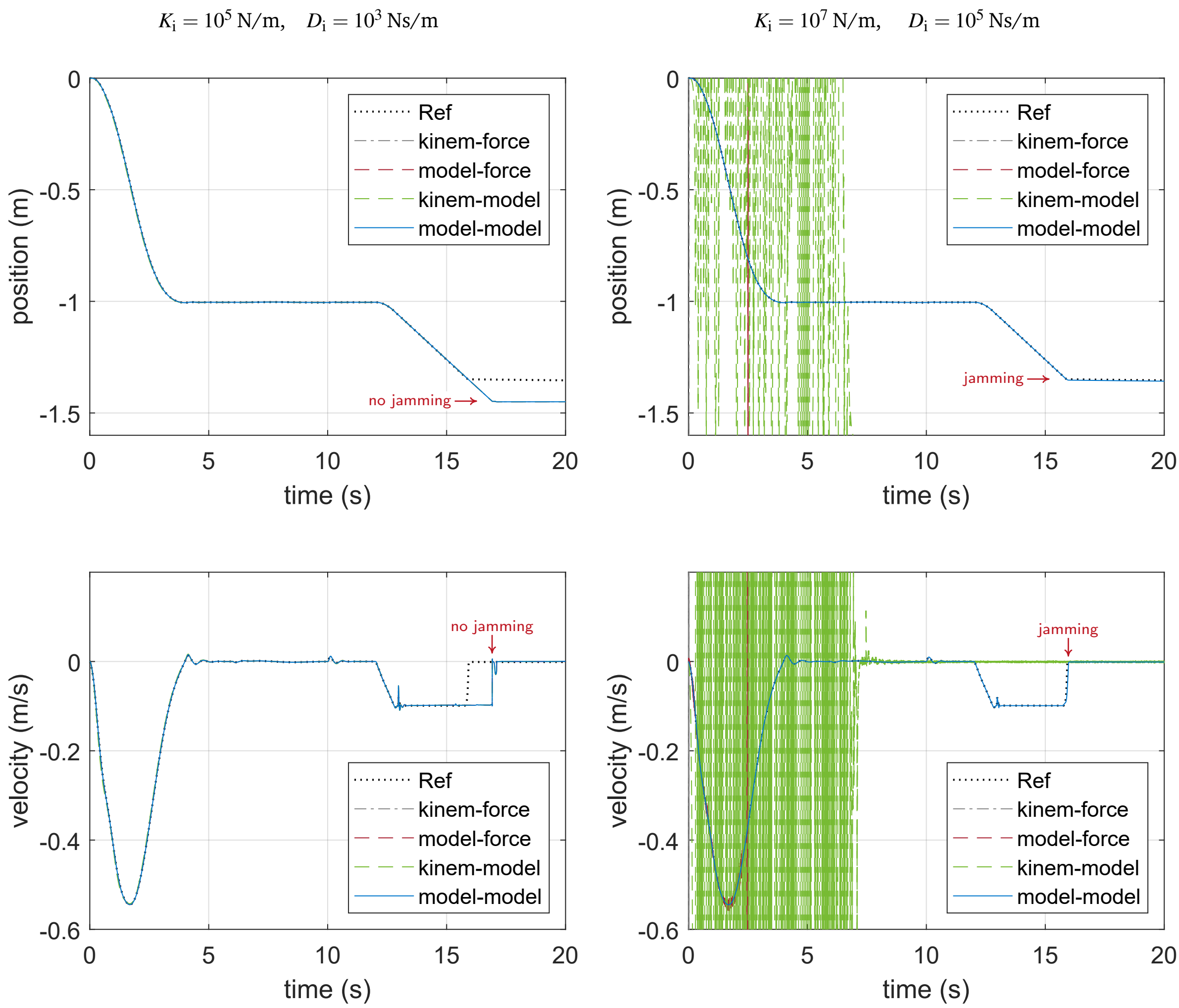

Figure 6: Position and velocity of the end effector during the insertion task for two sets of interface stiffness K_i and damping D_i values.

constraints. The bottle neck of this computation is the inverse matrix $\left(\hat{\mathbf{A}}_2\mathbf{M}_2^{-1}\hat{\mathbf{A}}_2^\text{T}\right)^{-1}$ in Eq. (20), the size of which is the number of active constraints (i.e., bilateral constraints, closed contacts, and static friction). This operation was optimized using a sparse-matrix Cholesky decomposition and solving for the columns of the right-hand matrix, so that the inverse matrix does not have to be calculated. The computational complexity appears to lie in between $O(n)$ and $O(n^2)$, and it shows some overhead for a small number of constraints.

The proposed method improved simulation accuracy and coupling stability. However, considering the properties of reduced model constant within the co-simulation macro step is a limitation of our method. Therefore, future work could focus on how changes in the subsystems affect the formulation of reduced models (e.g., contact detachment and stick-slip transitions). Moreover, improvements to the computational performance of the method and the scalability in

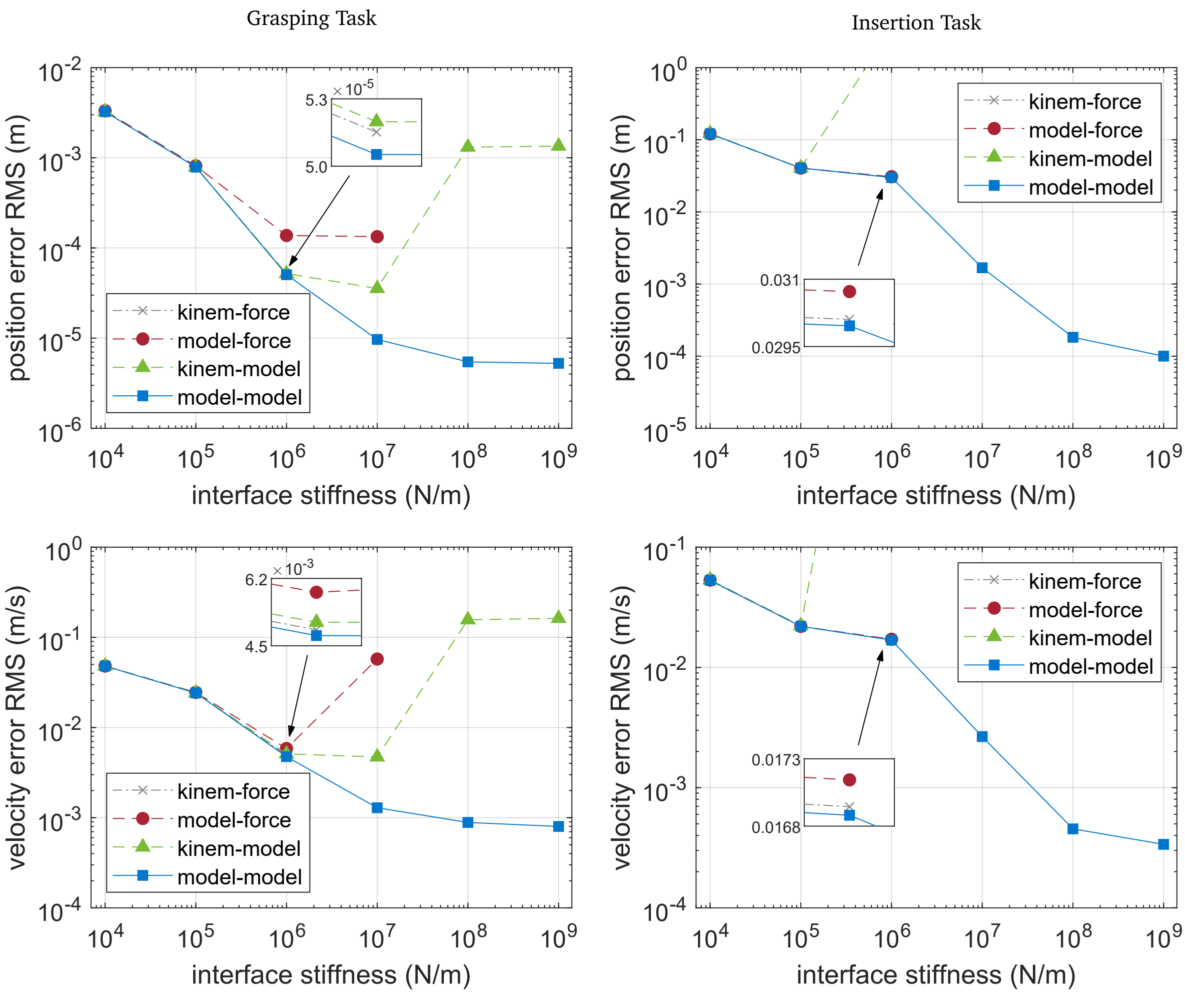

Figure 7: Root-mean-square (RMS) error of the position and velocity of the arm's end effector compared with the reference for different interface stiffness values K_i. Proportional damping $D_\mathrm{i}/K_\mathrm{i} = 10^{-2}\,\mathrm{s}$ was used in all the simulations, and angular stiffness and damping values were reduced by a factor of 10 compared to the linear ones.

large-scale simulations can also be of interest for future applications.

9 Conclusions

Different coupling methods for co-simulation of mechanical systems were discussed and analyzed. In addition, a framework for model-based coupling of subsystems with contact was put forward, where reduced-order models can be exchanged between subsystems. These models are formulated using effective mass and effective force terms, and can be combined and simulated with other mechanical systems. Numerical results of co-simulation setups of robotic systems performing challenging tasks such as grasping and insertion were used to illustrate the benefits

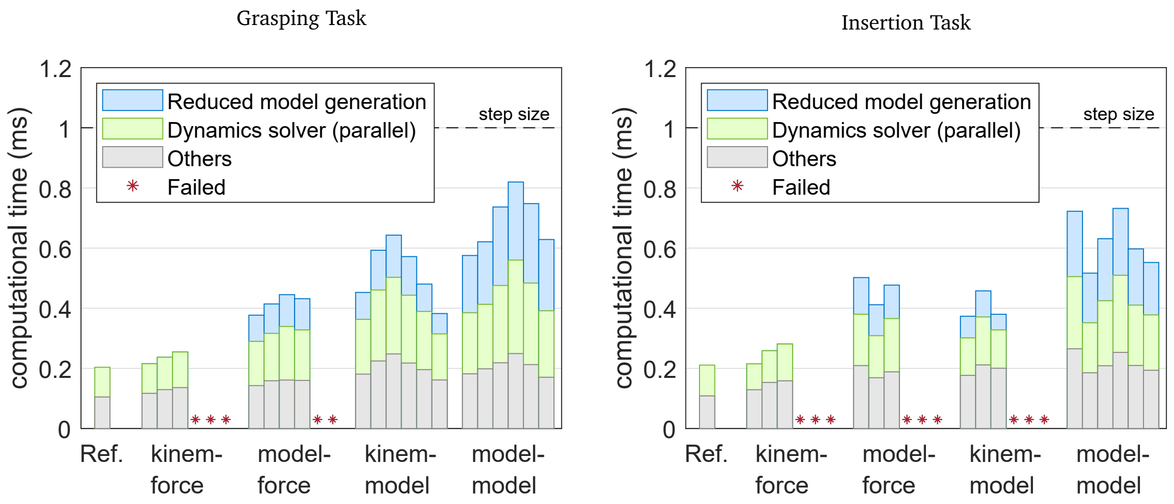

Figure 8: Average computational time (wall-clock) of co-simulation steps of size $h = 1\,\text{ms}$. Interface stiffness $K_\text{i} = \{10^4, 10^5, 10^6, 10^7, 10^8, 10^9\}\,\text{N/m}$ for each method, and the reference without co-simulation. The total time is divided into: reduced model generation time (calculation of effective mass and effective force terms), dynamics simulation time, and others (subsystem data exchange, collision detection, etc.). Results were obtained on an Intel Core i7-8750H CPU at 2.2 GHz with 16 GB of memory and 64-bit Windows 10.

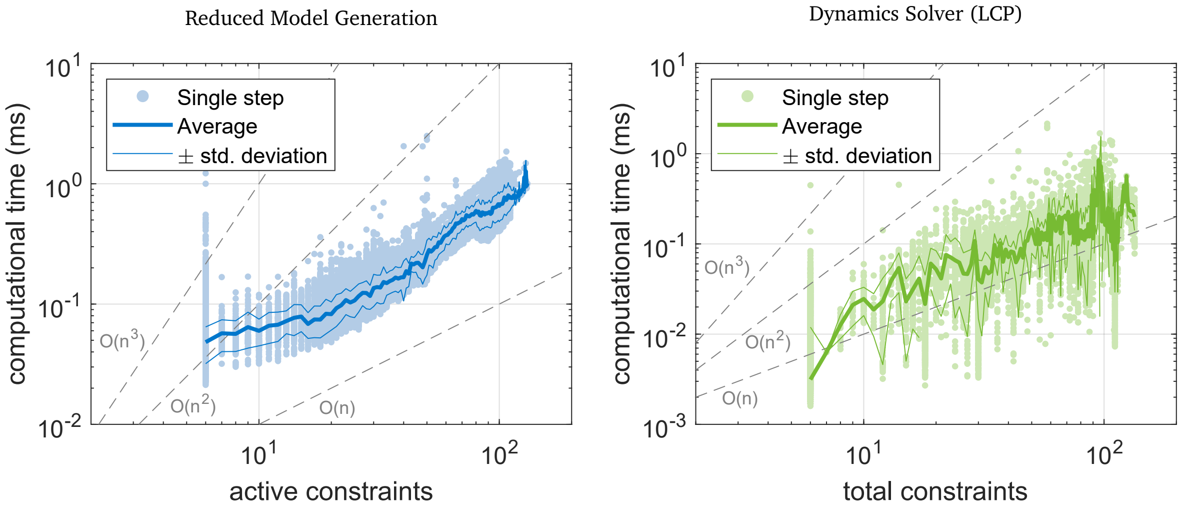

Figure 9: Computational time (wall-clock) to generate the reduced model in terms of the number of active constraints (left) and solve the system dynamics in terms of the total number of constraints (right) of subsystem 1 (plug with socket) in the insertion task example (step data from failed simulations are also included). Complexity-order lines are scaled and shown as reference only.

of the proposed method. Model-based coupling improved simulation stability and allowed for larger interface stiffness, which resulted in stronger coupling between subsystems and improved simulation accuracy.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and CM Labs Simulations, Inc. F. González was supported by the Ramón y Cajal program of the Ministry of Economy of Spain, contract no. RYC-2016-20222. The support is gratefully acknowledged.

References

[1] Acary, V., Brogliato, B.: Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics. Springer Science & Business Media (2008)

[2] Anitescu, M., Potra, F.A.: Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. Nonlinear Dynamics **14**(3), 231–247 (1997). DOI 10.1023/A:1008292328909

[3] Antunes, P., Magalhães, H., Ambrosio, J., Pombo, J., Costa, J.: A co-simulation approach to the wheel–rail contact with flexible railway track. Multibody System Dynamics **45**(2), 245–272 (2019)

[4] Arnold, M.: Stability of sequential modular time integration methods for coupled multibody system models. Journal of Computational and Nonlinear Dynamics **5**(3) (2010)

[5] Arnold, M., Burgermeister, B., Führer, C., Hippmann, G., Rill, G.: Numerical methods in vehicle system dynamics: state of the art and current developments. Vehicle System Dynamics **49**(7), 1159–1207 (2011). DOI 10.1080/00423114.2011.582953

[6] Benedikt, M., Watzenig, D., Hofer, A.: Modelling and analysis of the non-iterative coupling process for co-simulation. Mathematical and Computer Modelling of Dynamical Systems **19**(5), 451–470 (2013). DOI 10.1080/13873954.2013.784340

[7] Glocker, C.: Set-Valued Force Laws. Springer, Troy, New York, USA (2001)

[8] Júdice, J., Pires, F.: Basic-set algorithm for a generalized linear complementarity problem. Journal of optimization theory and applications **74**(3), 391–411 (1992). DOI 10.1007/BF00940317

[9] Júdice, J.J.: Algorithms for linear complementarity problems. Algorithms for continuous optimization **434**, 435–474 (1994). DOI 10.1007/978-94-009-0369-2_15

[10] Khatib, O.: Inertial properties in robotic manipulation: An object-level framework. The international journal of robotics research **14**(1), 19–36 (1995)

[11] Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. Multibody System Dynamics **4**(2), 107–127 (2000). DOI 10.1023/A:1009810318420

[12] Peiret, A., Andrews, S., Kövecses, J., Kry, P.G., Teichmann, M.: Schur complement-based substructuring of stiff multibody systems with contact. ACM Transactions on Graphics (TOG) **38**(5), 1–17 (2019)

[13] Peiret, A., González, F., Kövecses, J., Teichmann, M.: Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems. Mechanism and Machine Theory **127**, 52–72 (2018). DOI 10.1016/j.mechmachtheory.2018.04.016

[14] Peiret, A., González, F., Kövecses, J., Teichmann, M.: Co-Simulation of Multibody Systems With Contact Using Reduced Interface Models. Journal of Computational and Nonlinear Dynamics **15**(4) (2020). DOI 10.1115/1.4046052. 041001

[15] Rahikainen, J., González, F., Naya, M.Á.: An automated methodology to select functional co-simulation configurations. Multibody System Dynamics **48**(1), 79–103 (2020)

[16] Schweizer, B., Li, P., Lu, D.: Explicit and implicit cosimulation methods: Stability and convergence analysis for different solver coupling approaches. Journal of Computational and Nonlinear Dynamics **10**(5), 051,007,1–12 (2015). DOI 10.1115/1.4028503

[17] Schweizer, B., Li, P., Lu, D.: Co-simulation method for solver coupling with algebraic constraints incorporating relaxation techniques. Multibody System Dynamics **36**(1), 1–36 (2016). DOI 10.1007/s11044-015-9464-9

[18] Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. Archive of Applied Mechanics **84**(12), 1739–1769 (2014). DOI 10.1007/s00419-014-0883-5

[19] Schweizer, B., Lu, D.: Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. Multibody System Dynamics **34**(2), 129–161 (2015). DOI 10.1007/s11044-014-9422-y

[20] Sicklinger, S., Belsky, V., Engelmann, B., Elmqvist, H., Olsson, H., Wüchner, R., Bletzinger, K.U.: Interface jacobian-based co-simulation. International Journal for numerical methods in engineering **98**(6), 418–444 (2014)

[21] Stettinger, G., Horn, M., Benedikt, M., Zehetner, J.: Model-based coupling approach for non-iterative real-time co-simulation. In: 2014 European Control Conference (ECC), pp. 2084–2089. IEEE (2014)

[22] Stewart, D.E., Trinkle, J.C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. International Journal for Numerical Methods in Engineering **39**(15), 2673–2691 (1996). DOI 10.1002/(SICI)1097-0207(19960815)39: 15⟨2673::AID-NME972⟩3.0.CO;2-I

[23] Todorov, E.: Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6054–6061. IEEE (2014)

[24] Tomulik, P., Fraczek, J.: Simulation of multibody systems with the use of coupling techniques: a case study. Multibody System Dynamics **25**(2), 145–165 (2011)

[25] Trinkle, J.C., Pang, J.S., Sudarsky, S., Lo, G.: On dynamic multi-rigid-body contact problems with coulomb friction. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **77**(4), 267–279 (1997)

[26] Wang, J., Ma, Z.D., Hulbert, G.M.: A gluing algorithm for distributed simulation of multibody systems. Nonlinear dynamics **34**(1-2), 159–188 (2003)