On the cosimulation of multibody systems and hydraulic dynamics

Jarkko Rahikainen[1] · **Francisco González**[2] ·
Miguel Ángel Naya[2] · **Jussi Sopanen**[1] · **Aki Mikkola**[1]

Abstract The simulation of mechanical devices using multibody system dynamics (MBS) algorithms frequently requires the consideration of their interaction with components of a different physical nature, such as electronics, hydraulics, or thermodynamics. An increasingly popular way to perform this task is through co-simulation, that is, assigning a tailored formulation and solver to each subsystem in the application under study and then coupling their integration processes via the discrete-time exchange of coupling variables during runtime. Co-simulation makes it possible to deal with complex engineering applications in a modular and effective way. On the other hand, subsystem coupling can be carried out in a wide variety of ways, which brings about the need to select appropriate coupling schemes and simulation options to ensure that the numerical integration remains stable and accurate. In this work, the co-simulation of hydraulically actuated mechanical systems via noniterative, Jacobi-scheme co-simulation is addressed. The effect of selecting different co-simulation configuration options and parameters on the accuracy and stability of the numerical integration was assessed by means of representative numerical examples.

✉ J. Rahikainen
jarkko.rahikainen@lut.fi

F. González
f.gonzalez@udc.es

M.Á. Naya
miguel.naya@udc.es

J. Sopanen
jussi.sopanen@lut.fi

A. Mikkola
aki.mikkola@lut.fi

[1] Department of Mechanical Engineering, LUT University, Skinnarilankatu 34, Lappeenranta, Finland

[2] Mechanical Engineering Laboratory, University of A Coruña, Mendizábal s/n, Ferrol, Spain

1 Introduction

During the last decades, multibody-based simulation tools have established themselves in industry as they allow rapid testing of mechanical systems without costly prototyping. With the increase in computational power at affordable prices, these tools have also achieved the real-time limit outside the realm of academic discussions; see, for example, [1, 2]. However, in parallel with the maturity of this technology, the demand has risen for system-level simulation tools. Most engineering applications of industrial interest include not only mechanical components whose behavior can be described with conventional multibody system dynamics (MBS) formulations, but also elements with different physical properties, behavior, and time scale. These include control and electronics components, hydraulics circuits, and thermodynamic subsystems, among many others. When this is the case, the dynamics of the multibody components can no longer be considered on its own; instead, the interaction with the rest of elements in the engineering assembly must be appropriately described and taken into account in the simulation. For some of these applications, it is possible to develop *monolithic* formulations that are capable of describing the overall system dynamics with a common set of equations [3].

Monolithic methods are often accurate and efficient but require the modeling details of every subsystem to be accessible by the solution solver. This poses a problem in industrial applications in which at least some of the information about the system components must remain undisclosed to safeguard intellectual property rights. Moreover, in some applications the different physical behavior and time scales of the subsystems can make it challenging to develop a dynamics formulation able to represent all the phenomena involved. Assigning a particular solver to each subsystem, on the other hand, makes it possible to tailor its equations and implementation to the nature of the component that it is meant to represent. Although it would be, in theory, possible to build an all-encompassing simulation tool to suit all the engineering needs, in practice, this may not always be the case, and, in addition, the use of tailored modeling and solver tools for each engineering field is often advantageous. *Co-simulation* is an alternative to monolithic methods, which addresses these issues by assigning a solver to each subsystem in an application and then coupling their solution processes rather than developing a unified model of the whole system under study [4]. The subsystems exchange information through *coupling variables*, that is, input and output variables, at *communication points* in a discrete-time fashion. The result is a modular simulation environment, in which the implementation details of each subsystem remain hidden from the others. Additionally, co-simulation setups enable the distribution of the computational workload between several processing units [5, 6], even over network connections, and thus they can be used to improve the computational efficiency.

One of the main disadvantages of co-simulation with respect to its monolithic counterparts is the introduction of discontinuities and delays caused by the discrete-time nature of the communication interface [7]. These can eventually render the numerical integration process unstable in some cases, especially when noniterative co-simulation schemes are adopted [8]. Iterative co-simulation methods can be used to alleviate this problem; see, for example, [9, 10]; their use, however, becomes problematic in certain applications. This is the case of some real-time environments, in which the available time to carry out the computations is limited, and of systems that include components that cannot retake their integration steps, for example, physical components in hardware- or system-in-the-loop settings. In these cases, noniterative co-simulation schemes must be used.

The stability of noniterative co-simulation schemes can be improved in a number of ways. Polynomial extrapolation and approximation are among the most popular and simplest approaches [11–14]. The only information they require is the time history of the coupling

variables exchanged at the co-simulation interface; moreover, they are relatively easy to implement. Further analysis of the evolution of the coupling variables has led to the development of methods to improve stability based on the energy exchanges at the interface [15, 16] and to the determination of instantaneous frequencies of the subsystems in the setup [17]. The availability of additional information about the behavior and properties of the subsystems enables the use of stabilization techniques that can be particularized to match the needs of particular problems [18, 19]. It is also possible to estimate these properties from the information contained in the coupling variables to develop correction algorithms [20].

Co-simulation has been applied successfully to a wide range of applications that involve mechanical systems in multiphysics contexts, such as automotive [12], railway [21, 22], biomechanics [23], and heavy machinery [24] simulation. Hydraulically actuated mechanisms are a particularly interesting case of these, as they are present in a large number of industry and research applications. Several monolithic approaches to solve together the dynamics of mechanical and hydraulic components can be found in the literature [25–27]. Hydraulics and mechanics have also been coupled by means of co-simulation schemes [24, 25] and publications exist that deal with particular applications of this approach [28, 29]. However, the systematic study of the coupling method and the effect of selecting a given co-simulation scheme and set of coupling parameters on the efficiency and accuracy of the numerical integration has received comparatively less attention [30].

The objective of the present paper is to gain insight into the co-simulation of multibody and hydraulic dynamics when noniterative Jacobi schemes are used to couple the subsystems. Although there exist other noniterative schemes, such as Gauss–Seidel, in this study, we have chosen the Jacobi scheme for its efficiency potential. Moreover, explicit Jacobi schemes are often the option of choice in industrial applications with demanding computational requirements, such as those that couple physical and virtual components, that is, Hardware-in-the-Loop and System-in-the-Loop setups. To this end, we carry out a benchmark study. We compare the influence on the solution and coupling accuracy of two sets of coupling variables, two approaches to extrapolate the coupling variables, two multibody dynamics formulations with different internal integrators, and three solvers for the hydraulic system. We take the largest achievable communication step-size as an indicator of the simulation efficiency; in this work, we adjust its value to match the integration step-size of the slow subsystem, that is, the mechanical one. It can be expected that the computational overhead associated with information exchanges at communication points in practical applications will be reduced when the size of the communication step increases. On the other hand, longer step-sizes decrease the computational load required to integrate the multibody system dynamics. The maximum achievable step-size required to integrate the hydraulic subsystem is also touched upon, since it can also be expected to have an effect on system-level efficiency. To limit the number of considered variables, here we consider only systems with constant integration step-sizes and matching time grids.

We use two representative examples of hydraulically actuated mechanical systems to illustrate the effect of the above-mentioned co-simulation configuration options on the overall system behavior and coupling accuracy. The results showed that the accuracy of Jacobi-scheme co-simulation of mechanical and hydraulic systems can be enhanced by a proper selection of coupling variables and integration schemes.

2 Simulation of multibody and hydraulic dynamics

For the purposes of this study, we implemented a generic co-simulation manager and relevant subsystems. Here we only briefly introduce the multibody and hydraulic dynamics, whereas the co-simulation scheme and coupling accuracy evaluation are covered in more detail.

2.1 Multibody formalisms

A mechanical system can be described with a set of n generalized coordinates $\mathbf{q}$ subjected to m kinematic constraints $\mathbf{\Phi} = \mathbf{0}$, which in this work are assumed to be holonomic for simplicity. Without loss of generality, we assume that the generalized velocities are the derivatives with respect to time of the coordinates $\mathbf{q}$, that is, $\mathbf{v} = \dot{\mathbf{q}}$. The equations of motion can then be expressed as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{Q} + \mathbf{Q}_c, \tag{1a}$$

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

where $\mathbf{M}$ is the $n \times n$ mass matrix, $\mathbf{c}$ contains the Coriolis and centrifugal forces, and $\mathbf{Q}$ and $\mathbf{Q}_c$ are the applied and constraint forces, respectively. The generalized constraint reactions can be expressed as $\mathbf{Q}_c = -\mathbf{\Phi}_\mathbf{q}^T \boldsymbol{\lambda}$, where $\mathbf{\Phi}_\mathbf{q} = \partial\mathbf{\Phi}/\partial\mathbf{q}$ is the $m \times n$ Jacobian matrix of the constraints, and $\boldsymbol{\lambda}$ contains m Lagrange multipliers.

The dynamics in (1a), (1b) are described by a system of differential-algebraic equations (DAEs). Several algorithms have been proposed to perform the numerical solution of such sets of equations. One of them is the index-1 augmented Lagrangian algorithm (ALI1), introduced in [31] and further developed in [32]. This method enforces the satisfaction of kinematic constraints via an iterative algorithm, in which both the Lagrange multipliers and the system accelerations $\ddot{\mathbf{q}}$ are consecutively updated:

$$(\mathbf{M} + \mathbf{\Phi}_\mathbf{q}^T \alpha \mathbf{\Phi}_\mathbf{q})\ddot{\mathbf{q}} + \mathbf{\Phi}_\mathbf{q}^T \boldsymbol{\lambda}^*$$
$$= \mathbf{Q} - \mathbf{c} - \mathbf{\Phi}_\mathbf{q}^T \alpha (\dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_t + 2\xi\omega\dot{\mathbf{\Phi}} + \omega^2\mathbf{\Phi}). \tag{2a}$$

$$\boldsymbol{\lambda}_{j+1}^* = \boldsymbol{\lambda}_j^* + \alpha(\ddot{\mathbf{\Phi}} + 2\xi\omega\dot{\mathbf{\Phi}} + \omega^2\mathbf{\Phi}), \tag{2b}$$

where $\boldsymbol{\lambda}^*$ are the m modified Lagrange multipliers, and the subscript j stands for the iteration number. The modified Lagrange multipliers are made proportional via a penalty factor α to the violation of kinematic constraints at the configuration, velocity, and acceleration levels, following an approach similar to Baumgarte stabilization [33], in which ω and ξ are scalar parameters.

The ALI1 formalism can be used together with a given numerical integrator, in this study a semi-implicit forward Euler formula, following a fixed-point integration procedure to conduct the forward-dynamics simulation of the system motion. If the state of a mechanism at time-step n, $\mathbf{q}_n$ and $\dot{\mathbf{q}}_n$, is known, then Eqs. (2a), (2b) can be used to obtain the accelerations $\ddot{\mathbf{q}}_n$. These can be later integrated to obtain the positions and velocities in time-step $n + 1$.

On the other hand, replacing the fixed-point scheme with a Newton–Raphson solution approach can contribute to the improvement of the efficiency and stability of the solution process [34]. This requires the merging of the integration formulas, for example, the Newmark integrator family, and the dynamics equations, the latter of which now take the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_\mathbf{q}^T \alpha \mathbf{\Phi} + \mathbf{\Phi}_\mathbf{q}^T \boldsymbol{\lambda}^* = \mathbf{Q}. \tag{3}$$

Fig. 1 A volume with equally distributed pressure and effective bulk modulus
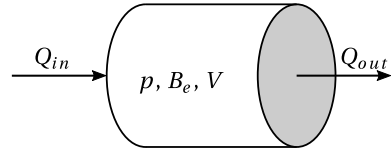
Introducing the equations of the integrator in the dynamics equations (3) and establishing the equilibrium at time-step $n + 1$, we obtain a system of nonlinear equations

$$\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}. \tag{4}$$

The solution of this nonlinear system of equations can be found through Newton–Raphson iteration:

$$\left[\frac{d\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})}{d\mathbf{q}} \right]_j \Delta\mathbf{q}_{j+1} = -\left[\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) \right]_j,$$
$$\mathbf{q}_{j+1} = \mathbf{q}_j + \Delta\mathbf{q}_{j+1}. \tag{5}$$

Mass-damping-stiffness orthogonal projections are used to ensure the satisfaction of kinematic constraints at the velocity and acceleration levels, thus improving the algorithm stability. Assuming that $\dot{\mathbf{q}}^*$ and $\ddot{\mathbf{q}}^*$ are the velocities and accelerations obtained after the Newton–Rhapson iteration has converged, these projections can be written for velocities as

$$\left[\frac{d\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})}{d\mathbf{q}} \right] \dot{\mathbf{q}} = \left[\mathbf{M} + \frac{\Delta t}{2}\mathbf{C} + \frac{\Delta t^2}{4}\mathbf{K} \right] \dot{\mathbf{q}}^* - \frac{\Delta t}{4} \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \alpha \mathbf{\Phi}_t \tag{6}$$

and for accelerations, as

$$\left[\frac{d\mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})}{d\mathbf{q}} \right] \ddot{\mathbf{q}} = \left[\mathbf{M} + \frac{\Delta t}{2}\mathbf{C} + \frac{\Delta t^2}{4}\mathbf{K} \right] \ddot{\mathbf{q}}^* - \frac{\Delta t}{4} \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \alpha \left(\dot{\mathbf{\Phi}}_{\mathbf{q}} \dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_t \right), \tag{7}$$

where $\mathbf{C}$ and $\mathbf{K}$ are the damping and stiffness contributions of the system. These steps represent the core of the index-3 augmented Lagrangian algorithm with projections (ALI3-P), described in [35] and extended to handle nonholonomic constraints in [36].

2.2 Hydraulic dynamics

The lumped fluid theory [37] is used in this study to model the dynamics of hydraulic actuators. According to this approach, the hydraulic circuit is divided into volumes V, in which the pressure p is assumed to be equally distributed. Each volume has sets of incoming and outgoing volume flows, Q_{in} and Q_{out}, respectively, and an effective bulk modulus B_e, which describes the flexibility of the oil and the containers, such as hoses and cylinders. The concept of the lumped fluid theory is illustrated in Fig. 1.

The pressure p can be solved from the first-order differential equation

$$\dot{p} = \frac{B_e}{V}(Q_{\mathrm{in}} - Q_{\mathrm{out}}), \tag{8}$$

Fig. 2 A generic co-simulation scheme

where V is the volume under investigation. The effective bulk modulus, in turn, takes the form

$$B_e = \left(B_o^{-1} + \sum_{l=1}^{n_c} \frac{V_l}{V B_l} \right)^{-1},$$ (9)

where B_o is the oil bulk modulus, n_c is the number of chambers in volume V, and V_l and B_l are the volume and bulk moduli of each chamber, respectively. The volume flows Q_{in} and Q_{out}, in turn, can originate from, for instance, cylinder movement or pressure difference over a valve, and therefore their equations are problem-dependent. Thus, these equations are presented with the case-example in Sect. 3.

2.3 Co-simulation setup

Figure 2 depicts a generic co-simulation scheme with, for simplicity, two subsystems and a generic co-simulation manager that couples the subsystems and orchestrates the co-simulation process. Each subsystem has its own internal set of states, which is integrated with method $\int_1$ and step-size h_1 in Subsystem 1 and integrator $\int_2$ and step-size h_2 in Subsystem 2. To couple their dynamics, the subsystems exchange information at communication steps of size H. Although, in general, all three step-sizes may be different, it is often practical to use the largest internal step-size as the communication step-size. It should also be noted that hereafter the notation of the subscripts h_1 and h_2 is for clarity particularized for each example.

In the used noniterative Jacobi scheme co-simulation, at each communication step the subsystems send outputs $\mathbf{y}_1$ and $\mathbf{y}_2$ and receive inputs $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively. During the communication step, both subsystems integrate their states independently. Since the updated values of inputs $\mathbf{u}_1$ and $\mathbf{u}_2$ are not available during this time, data extrapolation may be needed. For the purposes of this study, we implement a polynomial extrapolation and a reduced interface model [24]. The former method uses Lagrange polynomials, that is, it is oblivious to the actual system dynamics, whereas the latter makes a physics-based assumption on the mechanical system dynamics for the update of subsystem inputs, as will be explained next.

2.3.1 Co-simulation via interface models

In co-simulation setups, multibody systems are often coupled to other subsystems with faster dynamics, which need to be integrated with shorter step-sizes. When a multibody system $\mathcal{M}$ integrated with step-size $h_{\mathcal{M}}$ is coupled in a multirate Jacobi co-simulation setup to another subsystem $\mathcal{S}$ with faster dynamics, integrated with step-size $h_{\mathcal{S}} < h_{\mathcal{M}}$, the fast subsystem needs to anticipate its inputs between two consecutive communication points. This can be done by means of polynomial extrapolation. It is also possible to build an *interface model* (IM), to provide a prediction of the evolution of the dynamics of $\mathcal{M}$ between communication points [24]. Such an interface model expresses the dynamics of the multibody system $\mathcal{M}$ in

terms of the variables that describe its interface to $\mathcal{S}$. Let us assume that the dynamics at the interface can be described with a set of p generalized velocities $\mathbf{w}_i$ such that

$$\mathbf{w}_i = \mathbf{A}_i \dot{\mathbf{q}}, \tag{10}$$

where $\mathbf{A}_i$ is a $p \times n$ Jacobian matrix that makes it possible to express the interface velocities $\mathbf{w}_i$ in terms of the generalized velocities $\dot{\mathbf{q}}$ originally used to describe the multibody system. The matrix $\mathbf{A}_i$ also enables writing the interaction forces $\mathbf{Q}_i$ that couple the dynamics of $\mathcal{M}$ and $\mathcal{S}$ in terms of the local coupling forces at the interface $\boldsymbol{\lambda}_i$ as

$$\mathbf{Q}_i = \mathbf{A}_i^{\mathrm{T}} \boldsymbol{\lambda}_i. \tag{11}$$

Then the multibody dynamics can be expressed in terms of the velocities $\mathbf{w}_i$ that describe the interface as

$$\mathbf{M}_{\mathrm{eff}} \dot{\mathbf{w}}_i = \mathbf{Q}_{\mathrm{eff}} + \boldsymbol{\lambda}_i, \tag{12}$$

where

$$\mathbf{M}_{\mathrm{eff}} = \left(\mathbf{A}_i \left(\mathbf{I} - \mathbf{P}_c \right) \mathbf{M}^{-1} \mathbf{A}_i^{\mathrm{T}} \right)^{-1} \tag{13}$$

and

$$\mathbf{Q}_{\mathrm{eff}} = \mathbf{M}_{\mathrm{eff}} \left(\mathbf{A}_i \left(\mathbf{I} - \mathbf{P}_c \right) \mathbf{M}^{-1} \left(\mathbf{Q} - \mathbf{c} \right) + \dot{\mathbf{A}}_i \mathbf{w}_i + \mathbf{A}_i \mathbf{P}_c \dot{\mathbf{w}}_i \right), \tag{14}$$

and where $\mathbf{P}_c$ is the projector matrix

$$\mathbf{P}_c = \mathbf{M}^{-1} \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \left(\boldsymbol{\Phi}_{\mathbf{q}} \mathbf{M}^{-1} \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \right)^{-1} \boldsymbol{\Phi}_{\mathbf{q}}. \tag{15}$$

It should be noted that the present form of Eq. (15) relies on the mass matrix $\mathbf{M}$ being invertible, and therefore the method is limited to mechanical systems with a regular mass matrix.

The multibody dynamics provided by Eq. (12) can be used to obtain a physics-based prediction of the evolution of the inputs of the fast subsystem $\mathcal{S}$ between communication points in the way shown in Fig. 3.

The terms $\mathbf{M}_{\mathrm{eff}}$ and $\mathbf{Q}_{\mathrm{eff}}$ are evaluated at each communication point and sent to the interface model, which keeps them constant during the macrostep. These are used to formulate the dynamics of the interface model, which is integrated in a computationally inexpensive way with the same integration step-size of $\mathcal{S}$. This integration process is updated with the exchange of inputs between the IM and the faster subsystem $\mathcal{S}$. This way the IM provides $\mathcal{S}$ with a prediction of its inputs between communication points, which is based on the dynamics of the multibody system $\mathcal{M}$.

2.3.2 Systematic evaluation of co-simulation accuracy

In the benchmark process that will follow, a systematic method to evaluate the accuracy of the co-simulation is needed. Although the total energy of the system provides useful information about the system-level solution, it gives indication if the energy error is due to the decoupling of the problem or due to the internals of each subsystem. On the other hand, indicators, such as the RMS-value of position errors with respect to a certain reference solution, may be a poor indicator in some cases. For instance, a small phase shift in a high-frequency signal can appear as a large error, even if the resulting solution can be considered otherwise acceptable. For these reasons, this work uses a method based on the work presented

$h_{\mathcal{M}}$

Multibody System $\mathcal{M}$

$\mathbf{M}_{\text{eff}}, \mathbf{Q}_{\text{eff}}$ $\mathbf{Q}_{\text{i}}$

IM $\mathbf{M}_{\text{eff}} \dot{\mathbf{w}}_{\text{i}} = \mathbf{Q}_{\text{eff}} + \lambda_{\text{i}}$

$\mathbf{w}_{\text{i}}$ λ_{i} $\mathbf{w}_{\text{i}}$ λ_{i} $\mathbf{w}_{\text{i}}$ λ_{i} $\mathbf{w}_{\text{i}}$ λ_{i}

Faster subsystem $\mathcal{S}$

$h_{\mathcal{S}}$ $h_{\mathcal{S}}$ $h_{\mathcal{S}}$ t

Subsystem 1

t

$u_{k_1}(t_n)$ $y_{k_1}(t_n)$ $u_{k_1}(t_{n+1})$ $y_{k_1}(t_{n+1})$

Co-simulation manager

$y_{k_2}(t_n)$ $u_{k_2}(t_n)$ $y_{k_2}(t_{n+1})$ $u_{k_2}(t_{n+1})$

t

t_n t_{n+1}

Subsystem 2
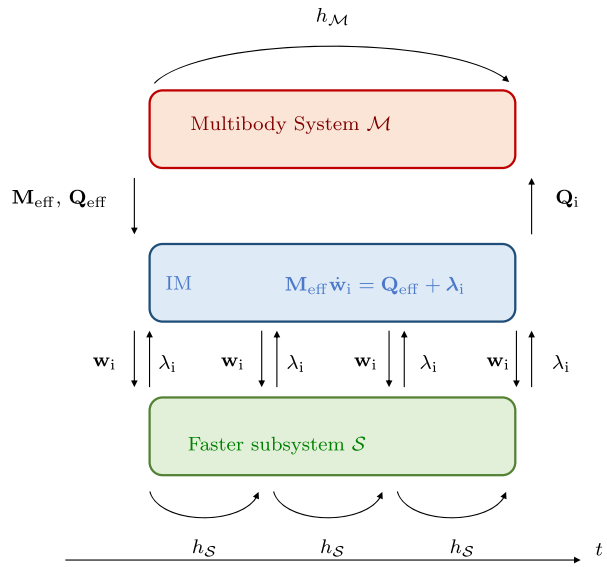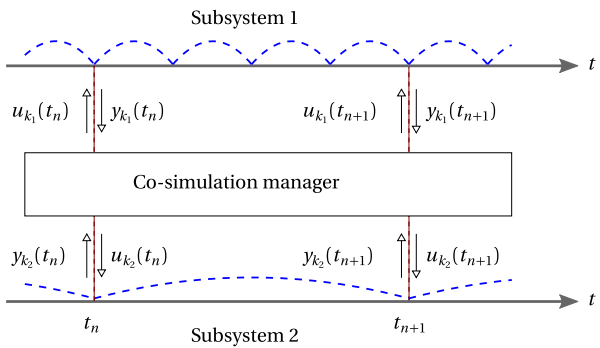
in [15], where the energy flows in the co-simulation interface are monitored via the concept of power bonds. This allows the error estimation at the co-simulation interface based on the input and output signals only, namely in terms of energy residuals. In contrast to [15], but similar to [16], the error indicator here can be considered a "generalized energy" and may not strictly represent energy in the physical sense.

To briefly introduce the key concepts related to the energy flow based error estimation, consider a Jacobi scheme co-simulation of two subsystems with scalar inputs and outputs between communication points t_n and t_{n+1}, as presented in Fig. 4. In the figure, at t_n, each subsystem sends and receives its input $u_k(t_n)$ and output $y_k(t_n)$, respectively, k being an index of power bond between the systems. After the time integration within each subsystem has taken place without additional communications, at time instant t_{n+1}, both subsystems send and receive $u_k(t_{n+1})$ and $y_k(t_{n+1})$, which are computed based on the previous inputs. With this information only, it is possible, as shown in [15], to estimate the introduced energy error in the co-simulation interface. It is important to note that, from hereafter, the generalized power and energy may not be power and energy in the physical sense if the product of inputs and outputs at each power bond does not correspond to an energy magnitude.

The concept of power bonds can now be applied to monitor the energy flow between the subsystems [15]. Applied to Fig. 4, this concept states that the energy transfer between Subsystems 1 and 2 via power ports k_1 and k_2 is called a power bond k. At communication time t, Subsystem 1 reports the total transmitted generalized power P_{k_1} as

$$P_{k_1}(t) = \tilde{u}_{k_1}(t) y_{k_1}(t), \tag{16}$$

where $\tilde{u}_{k_1}(t)$ is the extrapolated value of the previous input at the current communication time t, and $y_{k_1}(t)$ is the computed output. Subsystem 2, in turn, reports that the power it received from Subsystem 1 via the power port k_2 is

$$P_{k_2}(t) = \tilde{u}_{k_2}(t) y_{k_2}(t), \tag{17}$$

where $\tilde{u}_{k_2}(t)$ and $y_{k_2}(t)$ are the respective inputs and outputs.

Should the extrapolation within the subsystem be error-free, no energy would be lost or gained at the interface, and balance $-(P_{k_1} + P_{k_2}) = 0$ would hold. However, as the extrapolation error cannot be, in general, avoided, this balance will be violated, and therefore

$$-(P_{k_1} + P_{k_2}) \neq 0. \tag{18}$$

The above violation in the generalized power can now be exploited to monitor the co-simulation accuracy:

$$\delta P_k = -(P_{k_1} + P_{k_2}), \tag{19}$$

where δP_k is a generalized power residual of the power bond k, that is, the spuriously introduced power due to extrapolation errors. For multiple power bonds, this figure can be computed as

$$\delta P = -\tilde{\mathbf{u}}^{\mathrm{T}} \mathbf{y}, \tag{20}$$

where $\tilde{\mathbf{u}}$ and $\mathbf{y}$ are the vectors of extrapolated inputs and computed outputs, respectively. Should the number of inputs and outputs not be equal, as it will be the case with the used examples, only a subset of inputs or outputs is used to define a power bond. Following the notation introduced in Fig. 4, now we can write Eq. (20) as

$$\delta P_k(t_{n+1}) = -\tilde{\mathbf{u}}_k(t_{n+1})^{\mathrm{T}} \mathbf{y}_k(t_{n+1}), \tag{21}$$

which can be implemented in the co-simulation manager in a straightforward manner. The generalized residual energy is, in turn, defined piecewise between each communication instance with integral

$$\delta E_k(t_{n+1}) = \int_{t_n}^{t_{n+1}} \delta P_k(t) \, dt, \tag{22}$$

which now exactly defines the incorrectly added or dissipated energy that results from the extrapolation errors during the communication step $t_n \rightarrow t_{n+1}$. A numerical approximation can be used to compute Eq. (22) in the practical implementations. For $m = 0$, a rectangular rule can be used, for example,

$$\delta E_k(t_{n+1}) \approx \delta P_k(t_{n+1}) H(t_{n+1}), \tag{23}$$

where $H(t_{n+1})$ is the communication step size. For $m > 0$, higher-order quadrature rule should be used. Trapezoidal rule, which reads as

$$\delta E_k(t_{n+1}) \approx -\frac{H}{2}\big(\delta P_k(t_n) + \delta P_k(t_{n+1})\big), \tag{24}$$

is a straightforward solution with linear extrapolation. Although higher-order formulas, such as Simpson's rule, could provide a more accurate approximation, the outputs $\mathbf{y}_k$ are not known between communication instants, and therefore in this work, for extrapolation orders $m > 1$, we use the trapezoidal rule.

It should be noted that the residual energy in Eq. (22) defines the spurious energy added to the system or removed from it due to the decoupling of the original problem, regardless of the internal energy conservation in the subsystems. This means that the power sources or sinks in the subsystems are not visible in the residual, and therefore in this paper, two definitions exist for simulation accuracy. Firstly, there is the *coupling accuracy* defined in terms of Eq. (22) and its approximations. Secondly, there is *system-level accuracy*, which is defined in this paper by the energy balance of the entire system. While the former has an effect on the latter, since the decoupling error is added to the simulation error, the latter may not be visible in the former. These two definitions are jointly used to provide a comprehensive view to the accuracy of the simulation.

3 Examples

We implemented two nonlinear problems to evaluate co-simulation performance under different coupling approaches. Firstly, we used a planar model of a crane with a single actuator to examine the co-simulation with two interacting subsystems only. Secondly, we obtained a system with multiple actuators by attaching a second hydraulic cylinder to the crane. We modeled and integrated the dynamics of the two hydraulic actuators separately. In practice, hydraulic actuators are coupled through the dynamics of the mechanical system. In a Jacobi co-simulation setup, the integration of these two hydraulic subsystems proceeds separately between communication points, without information about the evolution of the other actuator. For this reason, keeping the co-simulation of the two-actuator model stable is more challenging than in the case of its single-actuator counterpart.

3.1 Multibody models

3.1.1 Single-actuator model

The first example is a two-link hydraulically actuated crane, shown in Fig. 5. A similar model was used as a test problem in [24] and [25].

The system has two degrees of freedom, one of which is controlled by a hydraulic actuator between the center point P of the first link and point B on the ground. The first link in the system is a rigid rod of length L with a homogeneously distributed mass m. Link 2 is a massless rod of length L_{h}. Point masses m_{p} and m_{h} are located at the tips of the first and second links, points Q and R, respectively. The set of generalized coordinates chosen to describe the example motion is $\mathbf{q} = \begin{bmatrix} x_{\mathsf{P}} \ y_{\mathsf{P}} \ \theta_1 \ x_{\mathsf{R}} \ y_{\mathsf{R}} \end{bmatrix}^{\mathrm{T}}$; three kinematic constraints $\boldsymbol{\Phi} = \mathbf{0}$ are used to enforce that the distance between points Q and R remains constant and that angle θ_1 is proportional to the x and y coordinates of point P. The interface velocity w_{i} in this

Fig. 5 A single-actuator crane
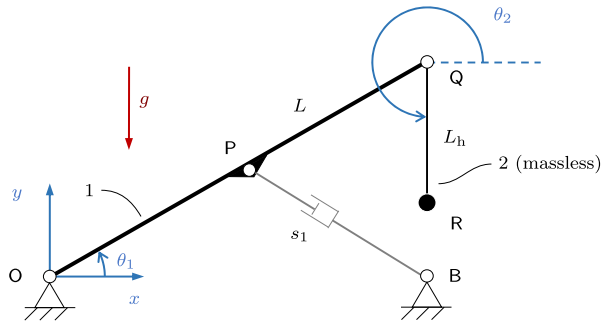
Table 1 Mechanical parameters of the single-actuated model

Parameter	Symbol	Value
Length of link 1	L	1.0 m
Length of link 2	L_h	0.5 m
Mass of link 1	m	200 kg
Point mass at Q	m_P	250 kg
Point mass at R	m_h	100 kg
Coordinates of fixed point B	(x_B, y_B)	$\left(\sqrt{3}/2, 0\right)$ m
Initial angle, link 1	$(\theta_1)_0$	$\pi/6$ rad
Initial angle, link 2	$(\theta_2)_0$	$3\pi/2$ rad
Gravity	g	-9.81 m/s^2

example corresponds to the actuator rate $\dot{s}_1$ and can be related to the generalized velocities as follows:

$$w_i = \dot{s}_1 = \left[\frac{x_P - x_B}{s_1} \quad \frac{y_P - y_B}{s_1} \quad 0\ 0\ 0 \right] \dot{\mathbf{q}} = \mathbf{A}_i \dot{\mathbf{q}}. \tag{25}$$

The force transmitted at the interface λ_i corresponds to the force exerted by the cylinder.

The physical properties of the single-actuator crane are summarized in Table 1.

3.1.2 Double-actuator model

The second example is a crane with two hydraulic actuators obtained after adding a second cylinder, identical to the previously existing one, between point P and the center point of link 2, as shown in Fig. 6. Such a system was used in [24] to assess the behavior of interface models in the co-simulation of multiply actuated systems.

The set of variables used to describe the system motion remains unchanged. However, a new component of the interface velocity is required to describe the interaction between the second actuator and the mechanical system. The rate $\dot{s}_2$ of the second cylinder is selected to this end:

$$\mathbf{w}_i = \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{x_P - x_B}{s_1} & \dfrac{y_P - y_B}{s_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{x_R}{4\,s_2} & \dfrac{y_R}{4\,s_2} \end{bmatrix} \dot{\mathbf{q}}. \tag{26}$$

Fig. 6 A double actuator crane
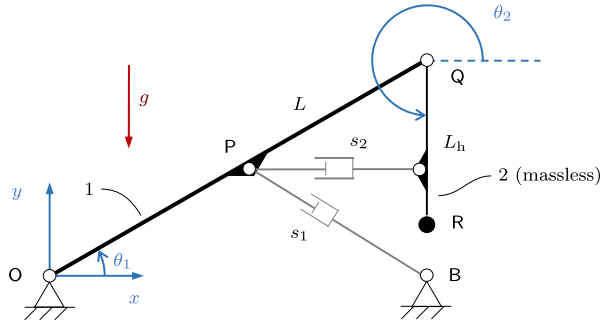
Fig. 7 A hydraulic circuit
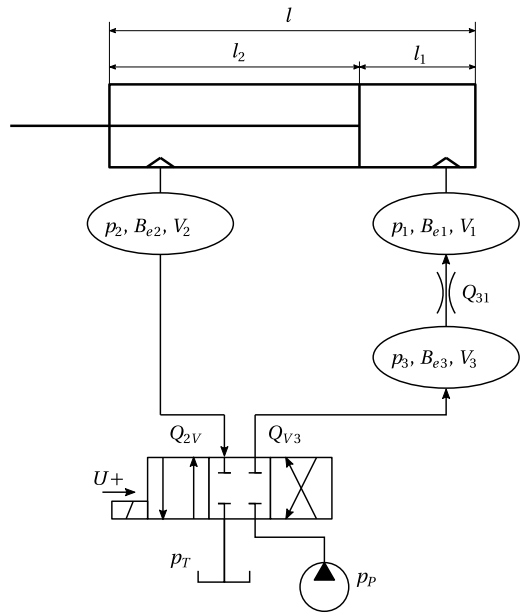
The mechanical properties of the system remain the same as those in the single-actuator model.

3.2 Hydraulic circuit

Figure 7 depicts the hydraulic actuator circuit used in both examples in Sect. 3. In the latter, this circuit is duplicated with slightly different parameters to add the second actuator to the system. The circuit includes a pump and a tank, which are assumed to have constant pressures p_P and p_T, respectively. Oil volumes flow from or to these sources via a 4/3 directional valve, which is controlled with a control signal U produced with a joystick, the positive direction of the signal being depicted in the figure. The circuit is split into three volumes, one between the directional and a throttle valve and one at each side of the piston head. Additionally, the figure depicts the total cylinder length l and lengths at rod and piston sides, l_2 and l_1, respectively. A full list of the used parameters is presented in Table 2.

Table 2 Parameters of the example circuit

Parameter	Symbol	Value
Length of the piston	l	0.3 m
Initial piston-side length	l_{1_0}	0.05 m
Initial rod-side length	l_{2_0}	0.25 m
End damper length	l_d	0.008 m
End damper spring coefficient	k_d	1×10^7 N/m
End damper damping coefficient	c_d	5×10^3 Ns/m
Initial actuator length	s_0	0.5 m
Fluid density	ρ	850 kg/m^3
Oil bulk modulus	B_o	1500 MPa
Cylinder bulk modulus	B_c	31500 MPa
Hose bulk modulus	B_h	150 MPa
Volume of hose 1	V_{h1}	3.14×10^{-5} m^3
Volume of hose 2	V_{h2}	7.85×10^{-5} m^3
Volume of hose 3	V_{h3}	7.85×10^{-7} m^3
Diameter of piston	d_1	80 mm
Diameter of piston rod	d_2	35 mm
Tank pressure	p_T	0.1 MPa
Pump pressure	p_P	7.6 MPa
Cylinder efficiency	η	0.88
Throttle discharge coefficient	C_d	0.8
Semiempirical coefficient	C_v	1.0690×10^{-8} m^3/s$\sqrt{\text{Pa}}$
$-45°$phase shift frequency	f_{-45}	35 Hz

The hydraulic circuit has three volumes, and therefore applying Eq. (8) yields three first-order differential equations, which describe the system dynamics:

$$\dot{p}_1 = \frac{B_{e1}}{V_1}(Q_{31} - A_1\dot{s}), \tag{27}$$

$$\dot{p}_2 = \frac{B_{e2}}{V_2}(A_2\dot{s} - Q_{2V}), \tag{28}$$

$$\dot{p}_3 = \frac{B_{e3}}{V_3}(Q_{V3} - Q_{31}), \tag{29}$$

where A_1 and A_2 are piston and rod side areas in the cylinder, and $\dot{s}$ is the actuator rate. The volume flows over the valves, Q_{2V}, Q_{V3}, and Q_{31}, are computed with the semiempirical model [38], and their positive directions are depicted in Fig. 7. The expression for the volume flow over the throttle valve can be written as

$$Q_{31} = \begin{cases} C_t\sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_t\sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{30}$$

where C_t is the semiempirical coefficient, and Δp is the pressure difference over the valve. In the case of a simple throttle valve the semiempirical coefficient is computed as $C_t =$

$C_d A_t \sqrt{\frac{2}{\rho}}$, where C_d is the discharge coefficient, A_t is the throttle area, and ρ is the oil density. The volume flows Q_{2V} and Q_{V3} over the directional valve take the form

$$Q = \begin{cases} C_v U \sqrt{|\Delta p|}, & \Delta p > 0, \\ 0, & \Delta p = 0, \\ -C_v U \sqrt{|\Delta p|}, & \Delta p < 0, \end{cases} \tag{31}$$

where C_v is the semiempirical coefficient computed based on one operational point of the valve, and U is the control signal computed as

$$\dot{U} = \frac{U_{\text{ref}} - U}{\tau}, \tag{32}$$

where U_{ref} the reference, in this case from the joystick, U is the achieved position, and $\tau = (2\pi f_{-45})^{-1}$ is the time constant computed based on the $-45°$ phase shift frequency.

The force produced by the cylinder can be computed as follows:

$$F_s = p_1 A_1 - p_2 A_2 - c\dot{s} - F_d, \tag{33}$$

where c is a viscous coefficient of 1×10^5 Ns/m, and F_d is the end damper force computed in this case as

$$F_d = \begin{cases} k_d(l_d - l_1) - c_d \dot{s}, & l_1 \leq l_d, \\ -k_d(l_d - l_2) - c_d \dot{s}, & l_2 \leq l_d, \\ 0 & \text{otherwise,} \end{cases} \tag{34}$$

where k_d and c_d are, respectively, the spring and damping coefficients of the dampers, and l_d is the damper length. Note that the cylinder lengths l_1 and l_2, as depicted in Fig. 7, include the dampers.

3.3 Co-simulation setup

The described systems are coupled by the co-simulation manager via a noniterative Jacobi scheme. The mechanical system sends at each communication point the actuator length s, velocity $\dot{s}$, and acceleration $\ddot{s}$ to the manager. The hydraulic systems, in turn, send either the cylinder force F_s or the pressures $\mathbf{p} = [p_1 \; p_2 \; p_3]$ to the interface. In the latter option the cylinder force is computed within the multibody system; accordingly, information about the properties of the hydraulic system needs to be available to the multibody subsystem, so that it can evaluate Eq. (33) to obtain the cylinder force. These two options are hereafter referred to as velocity–force and velocity–pressure couplings, and both can be used with and without the reduced interface model. Polynomial extrapolation, based on the Lagrange polynomials, is used on the inputs of the hydraulic system when the IM is not used.

Regarding time integration, the mechanical system is solved with the index-1 and index-3 augmented Lagrangian algorithms described in Sect. 2.1 and shares the constant communication step-size of the co-simulation manager, that is, $h_m = H$. The ALI1 formulation is integrated with a semi-implicit forward Euler formula, and the ALI3-P with the trapezoidal rule integrator. The hydraulic system, in turn, has a smaller internal step-size, that is, $h_h < H$, which forms a matching grid with the communication instances. To evaluate the effect of the internal integration scheme of the hydraulic subsystem, three different integrators are implemented for experimenting, namely the forward Euler formula (labeled Euler) as a

Table 3 Configuration variables

Mech	Hyd	Coupling variables	Extrapolation
ALI1	Euler	$s, \dot{s}, F_s$	Order m
ALI3-P	RK4	$s, \dot{s}, \mathbf{p}$	IM
	TR		

Fig. 8 Simulated work cycle
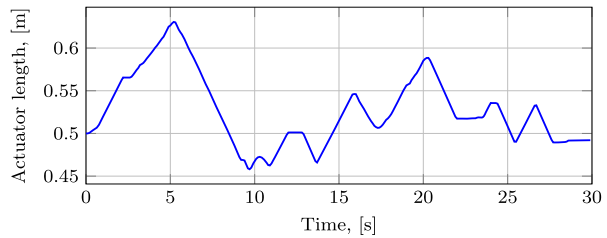
single-step explicit integrator, the fourth-order Runge–Kutta method (RK4) as a multistep explicit integrator, and the trapezoidal rule (TR) as an implicit integrator. Fixed step-sizes are used during the integration process. These configurations are summarized in Table 3.

4 Results

The given examples are used in the benchmark to gain insight into the noniterative Jacobi-scheme co-simulation of multibody and hydraulic dynamics. The focus of the benchmark is on the influence of the configuration on the achievable communication step-size, since it is known to have a significant effect on co-simulation accuracy and efficiency [12, 18]. To this end, the internal step-size h_h of hydraulics is locked to a value, in this case 0.1 ms, small enough to obtain a stable solution, whereas H is varied under each tested co-simulation configuration. The generalized residual energy at the end of the simulation is taken as an indicator of the coupling accuracy, whereas the overall solution accuracy is measured by the energy balance of the entire system. In addition to the study of the achievable communication step-size with a predetermined value of h_h, here the effect of the internal integrator of the hydraulics subsystem on the achievable internal step-size is also briefly touched upon. A monolithic solution, which follows the integration scheme shown in [26], is used to produce a set of reference results with the ALI3-P formulation. As the case examples are implemented in a Matlab environment and the used implementation does not produce comparable computation times, here CPU-times are omitted.

4.1 Single actuator model

For the first studied case, that is, the single actuator crane in Sect. 3.1.2, a reference signal U is produced for the directional valve with a joystick. The resultant actuator length that corresponds to this input is depicted in Fig. 8. During the work cycle, the actuator hits both end dampers, the models of which are similar to contact models, which can be assumed to be challenging for the integrators. Figure 8 shows the reference result obtained with a monolithic simulation at 1 ms step size. This reference solution is compared to others obtained with Jacobi-scheme co-simulation using different coupling strategies.

The effect of the communication step-size H, used integrators $\int_m$ and $\int_h$, and extrapolation methods, that is, $m = 2$ and IM, on the coupling accuracy are shown in Figs. 9a–9f, where the generalized energy has been normalized so that the configuration of ALI1 with second-order extrapolation and forward Euler integrator for hydraulics at $h_h = 1$ ms internal step size equals zero. It must be noted that the generalized energy residuals are directly comparable when the coupling variables are the same, that is, within left- and right-hand side columns of figures, and in both cases, the lower energy residual indicates a better solution. The achievable step-size is taken to be the segment of the curve where the accumulated energy residual is relatively low, almost linearly dependent on the communication step size H. When the curve "takes off", accuracy of the coupling and therefore the simulation can be assumed to be lost.

As Fig. 9 shows, the index-1 formulation can achieve significantly larger communication step-sizes with a similar level of coupling error introduced at the co-simulation interface, when compared to the index-3 formulation. The lower energy residual likely results from the explicit integrator used with the ALI1 formulation, which eliminates the need for signal extrapolation, and thus is indifferent to the co-simulation scheme. However, as Fig. 10 shows, in turn, the index-1 solution introduces losses in the total system energy as the simulation progresses, whereas the index-3 solution maintains system-level accuracy until the stability of the solution is lost. This difference may stem from the dissipative terms in the ALI1 formulation and, in addition, from the used explicit integrator. This demonstrates the fact that the coupling accuracy alone cannot be used to fully determine if the solution for the system is accurate, even if it can be indicative of it, as is the case with the ALI1 solutions. However, on the other hand, the coupling accuracy can be computed even if the system level energy balance is not known and can be used, as Fig. 9 shows, as an indirect indicator of solution stability.

Regarding the use of the different coupling variables, the use of velocity pressure coupling seems to yield more accurate coupling and stable results, as Fig. 9 shows. This is true for both implemented formulations for the mechanical system, although in this case the index-1 formulation seems to benefit more from the use of pressures in the coupling, especially when the reduced interface model is used. This demonstrated that difference originates from the hydraulic force description in Eq. (33), i.e., $F_s = p_1 A_1 - p_2 A_2 - c\dot{s} - F_d$, which couples the mechanical and hydraulic states. If velocity-force coupling is used, then this equation is evaluated within the hydraulic subsystem, and for this reason, the actuator rate $\dot{s}$ is that extrapolated from the previous time step. This causes a direct feed-through, that is, the output of the hydraulic subsystem directly depends on its inputs. It has been shown in [8, 39] that algebraic loops caused by direct feed-through are detrimental for the stability of the cosimulated solution. Conversely, if a velocity–pressure coupling is used, then Eq. (33) is evaluated within the multibody system with the integrated value of $\dot{s}$ and pressures $\mathbf{p}$ obtained from the previous time step, which removes the direct feed-through and therefore, as demonstrated here, yields a more stable simulation.

As demonstrated in the literature, the used extrapolation method has a significant effect on co-simulation accuracy [12]. For the purposes of this study, polynomial extrapolation is used to compare to the interface model [24], which allows us to extrapolate the data from the mechanical system based on the system dynamics. The order of polynomial extrapolation was decided based on experiments with a configuration of ALI1, Euler, and $h_h = 0.1$ ms. As the results show, with the single-actuator model, polynomial extrapolation is performed in most cases better than the reduced interface model, both in the sense of solution and coupling accuracy and achievable communication step-size. Figure 11 exemplifies the difference in results between the polynomial extrapolation and reduced interface model. As we can see,
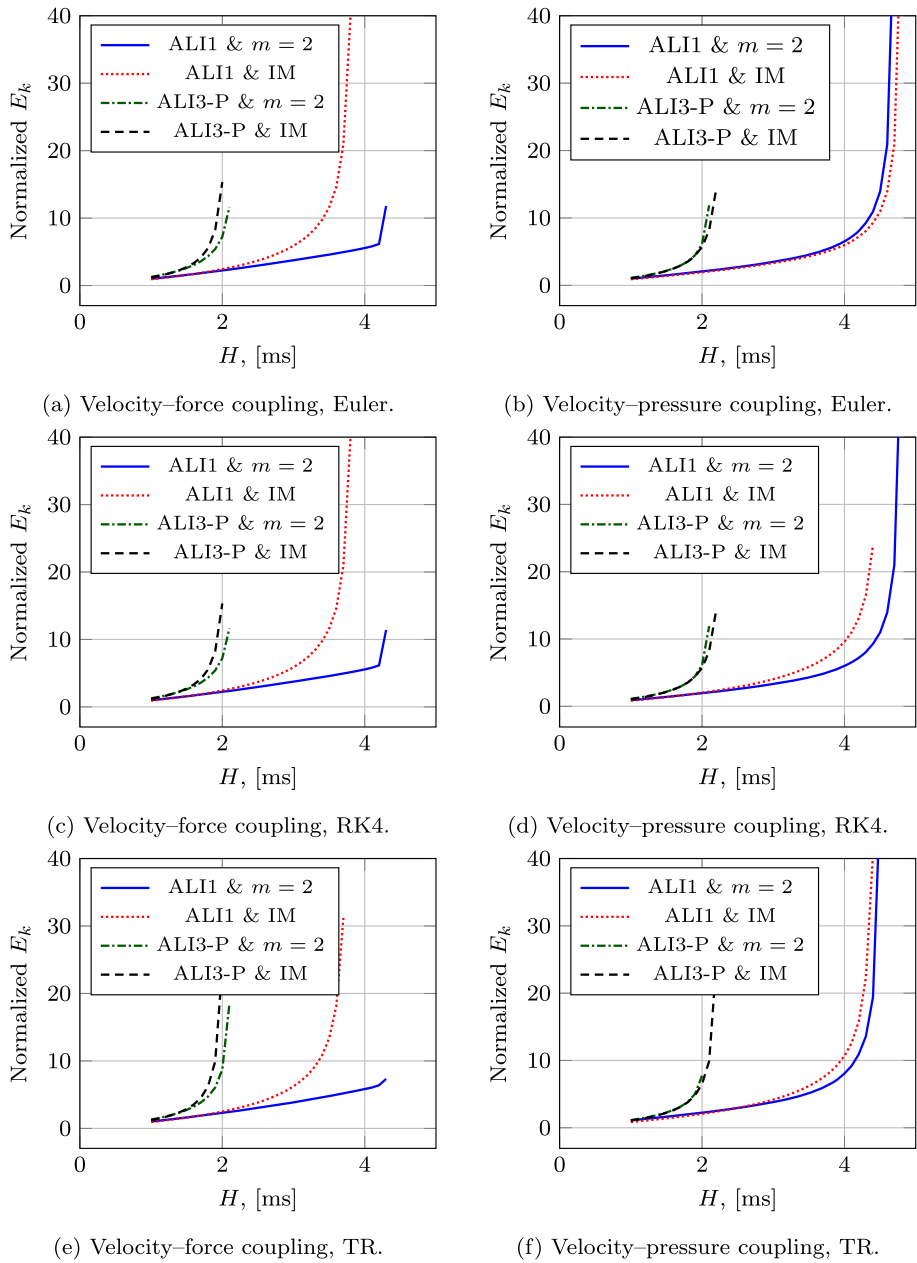
Fig. 9 Effect of the communication step-size H on coupling accuracy with different hydraulic integrators and coupling variables; single-actuator model

use of the reduced interface model may lead to degraded velocity level solution, whereas smaller deviations can be seen at the configuration level. This behavior also explains the increase in the residual energies seen in Fig. 9a.

Fig. 10 Energy balance with Euler integrator at $h_{hyd} = 0.1$ ms, polynomial extrapolation, and velocity-force coupling; single-actuator model
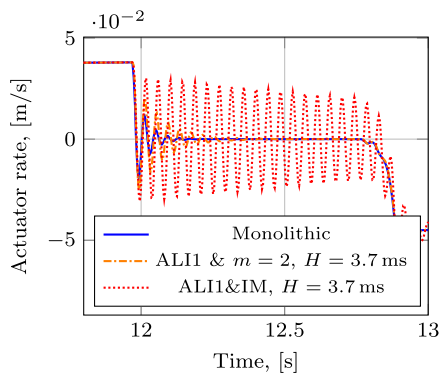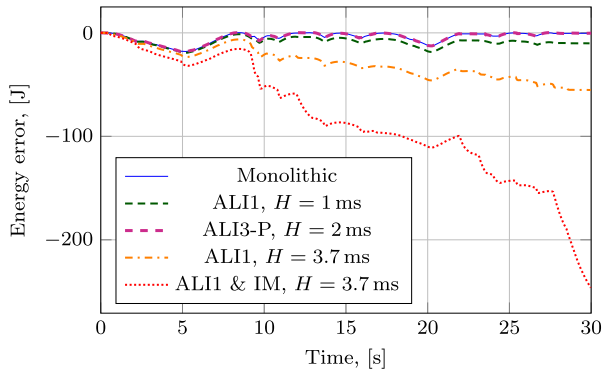
(a) Actuator rate.
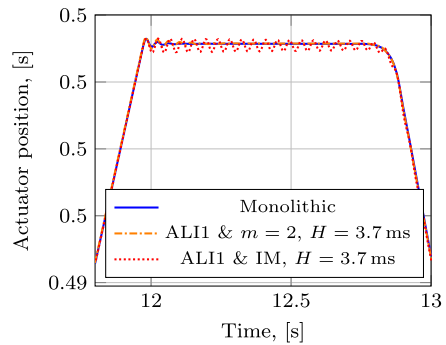
(b) Actuator length.

Fig. 11 Polynomial extrapolation vs. IM; single-actuator model

Figures 12 and 13 summarize the effect of the internal configuration of the fast subsystem on co-simulation accuracy. As already demonstrated in Fig. 9, the internal integrator of the fast subsystem, hydraulics in this case, in turn, seems to have only little impact on the achievable communication step-size H, should the internal step h_h be sufficiently small to ensure a stable integration. For a clearer demonstration, in Figs. 12a–12b the normalized energy residuals with different integrators for the hydraulic subsystem are presented when the rest of the system configuration is kept constant. This suggests that changing the hydraulic integrator may not allow larger communication step-sizes to be taken, should the internal step-size be small enough for the used integrator to produce an accurate solution for the system. Regarding the possibility to increase the internal step-size h_h, in turn, as demonstrated in Figs. 13a–13b, it seems to be possible only when H is sufficiently small. Notice how the solution with the trapezoidal rule integrator at $H = 4$ ms becomes unstable, whereas with $H = 1$ ms a stable solution can be found. The solution accuracy behaves in a similar manner. At $H = 1$ ms the solutions with $h_h = 0.1$ ms and $h_h = 0.5$ ms are closer to the monolithic solution than with $H = 4$ ms. This shows that in the studied case, there is a coupling between achievable H and the used h_h. This further indicates that depending on the location of the computational bottleneck of the co-simulation setup, it may be possible to optimize the overall efficiency by tuning step-sizes in fast and slow subsystems together.
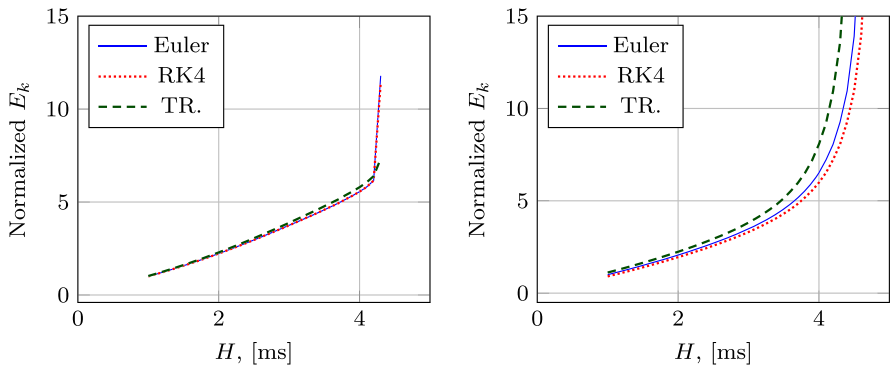
(a) ALI1 formulation, velocity–force coupling, and $h_h = 0.1\,\mathrm{ms}$.

(b) ALI1 formulation, velocity–pressure coupling, and $h_h = 0.1\,\mathrm{ms}$.

Fig. 12 Effect of the hydraulics integrator on the coupling accuracy; single-actuator model
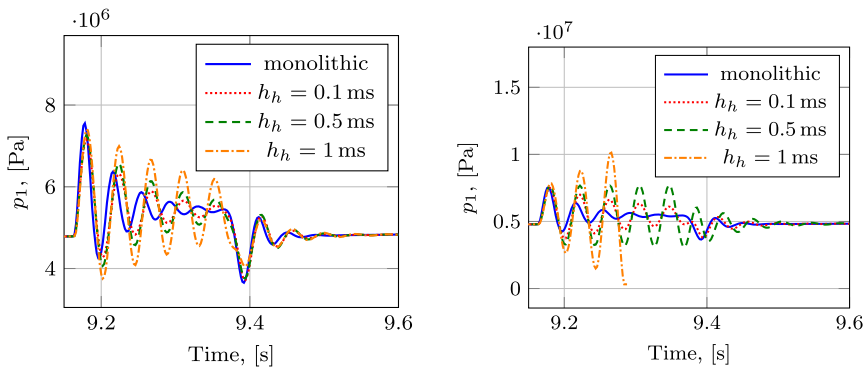
(a) ALI1, trapezoidal rule integration, velocity–force coupling, and $H = 1\,\mathrm{ms}$.

(b) ALI1, trapezoidal rule integration, velocity–force coupling, and $H = 4\,\mathrm{ms}$.

Fig. 13 Effect of the step size to the solution accuracy; single-actuator model

4.2 Double actuator model

Similarly to the first case, for the double-actuator model, reference signals for the directional valves of the hydraulic circuits are produced with a joystick. Figure 14 shows the resultant actuator lengths obtained with a monolithic simulation of 1 ms step-size. Although the system still seems fairly simple, the addition of the second hydraulic circuit as a new subsystem introduces an indirect coupling between the cylinders via the mechanical system, which from the co-simulation point of view, is significantly more challenging than the single-actuator case. The problem is solved again with the Jacobi-scheme co-simulation with different coupling strategies, as described before. To limit the number of considered variables, the same settings are used for both hydraulic subsystems.

Figure 15 shows the resultant generalized energy residuals, normalized in a similar manner to the previous case, that is, the ALI1 formulation with forward Euler integrator for the hydraulics, and first-order extrapolation at $H = 1\,\mathrm{ms}$ is set to one in the chart. As done with

Fig. 14 Simulated work cycle with the double-actuator model
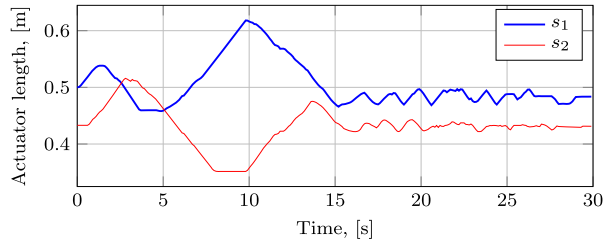
the single-actuator model, the internal step-size for both hydraulic subsystems has been kept at a constant of $h_h = 0.1$ ms in all cases, whereas other configuration parameters are varied.

As can be seen from Fig. 15, the achieved communication step-size H is, in general, smaller than those shown in Fig. 9 for the single-actuator model. This result can be expected, as the previously explained indirect coupling of the hydraulic subsystems makes this case more challenging for the used noniterative Jacobi scheme. An exception to the rule is the reduced interface model combined with the ALI1 formulation, which seems to be able to achieve communication step-sizes close to those of the single-actuator model without loss of stability. This result, in turn, agrees to the original proposal of the method [24] and, in addition, shows that the method seems to be viable with stiffer hydraulic models. The explanation behind the large communication step-sizes achieved with IM lies in its capability to maintain the coupling of the hydraulic subsystems between the communication instants.

Otherwise, the results follow the trends set by the single-actuator case. A clear difference can be seen in the results between the velocity–force and velocity–pressure couplings, the latter being able to achieve larger communication step-sizes. In this case, however, the benefit of the velocity–pressure coupling is more clear than in the single-actuator case, and it suggests that in more complex cases the elimination of the direct feed-through may be advisable in the used co-simulation scheme. The effect of the internal solver of the fast subsystem in this case, as Fig. 15 shows, also has little effect on the achievable communication step-size. This is again demonstrated in Figs. 16 and 17, where the generalized energy residuals are compared between the different subsystem integrators. The relation between the achievable communication step-size H and the internal step-size h_h is also exemplified. As we can see, should the communication step-size be small enough, the trapezoidal rule integrator can still achieve 1 ms step-sizes without losing stability, although with the increase in the communication step-size this property is quickly lost.

5 Conclusions

In this work, we presented a benchmark study to gain insight into noniterative Jacobi-scheme co-simulation of multibody and hydraulic dynamics. Two nonlinear examples are used for this purpose, with different approaches to the integration procedures and to the coupling. Their effects on the coupling and system accuracy have been investigated, and the key conclusions can be summarized as follows:

– The selection of variables for the information exchange can have a significant effect on the achievable communication step-size, since they may introduce or remove direct feed-through in the coupling scheme. In the light of the presented results, it seems that the use of the state variables as subsystem outputs in the coupling should be recommended.
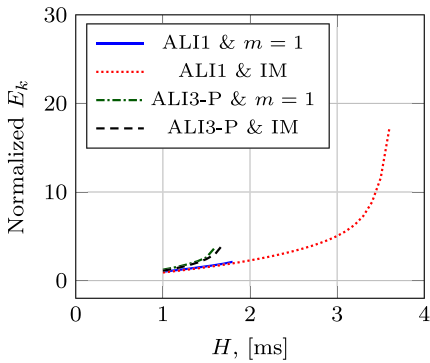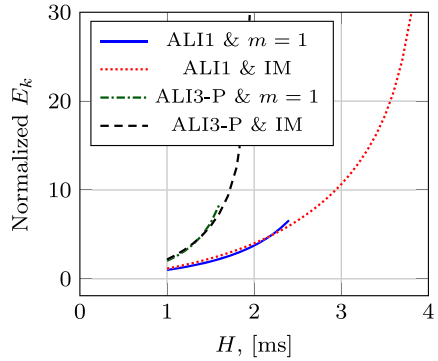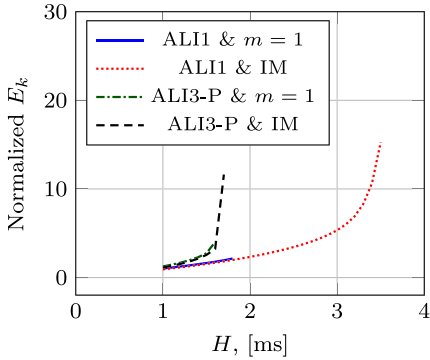
(a) Velocity–force coupling, Euler.
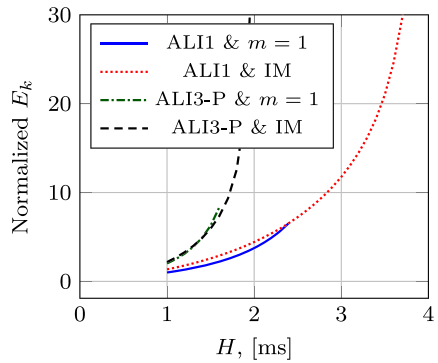
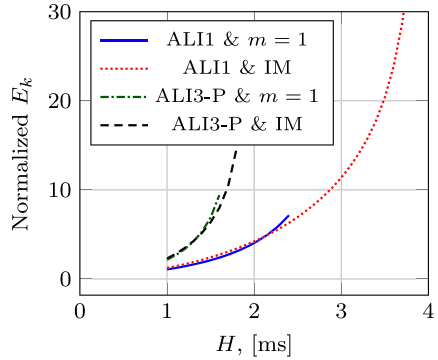(b) Velocity–pressure coupling, Euler.
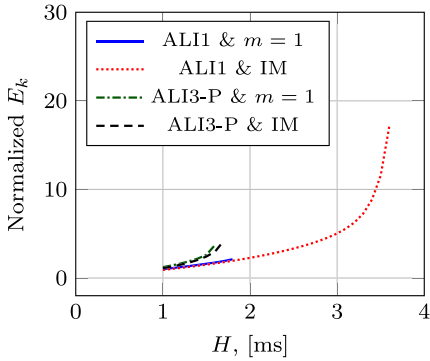
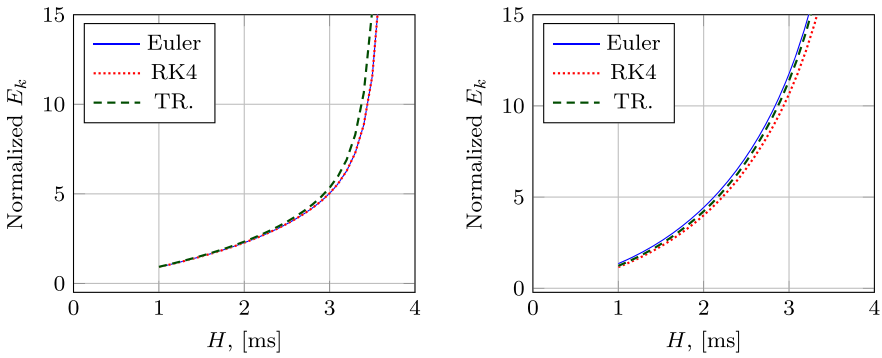(c) Velocity–force coupling, RK4.

(d) Velocity–pressure coupling, RK4.

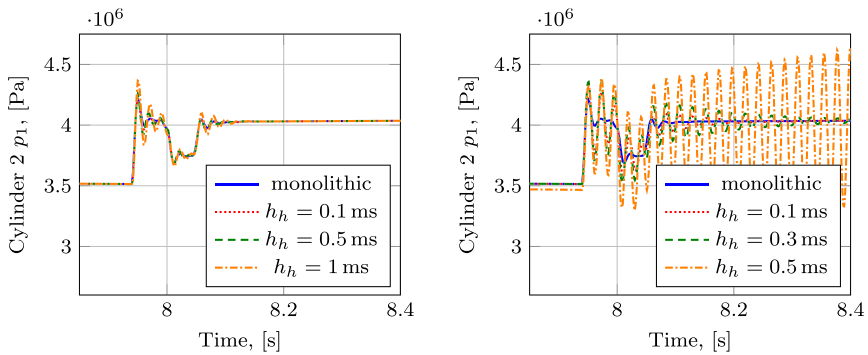(e) Velocity–force coupling, TR.

(f) Velocity–pressure coupling, TR.

Fig. 15 Effect of the communication step-size H on the coupling accuracy with different hydraulic integrators and coupling variables; double-actuator model

– The used multibody algorithm, and especially the used integrator, can have a significant effect on the achievable communication step-size and also on the accuracy of both the solution and the coupling itself. However, an accurate system level solution may not be the most stable one and vice versa, as the results show.

(a) ALI1 formulation, velocity–force coupling, and $h_h = 0.1$ ms. (b) ALI1 formulation, velocity–pressure coupling, and $h_h = 0.1$ ms.

Fig. 16 Effect of the hydraulics integrator on the coupling accuracy; double-actuator model

(a) ALI1, trapezoidal rule integration, velocity−force coupling, and $H = 1$ ms. (b) ALI1, trapezoidal rule integration, velocity−force coupling, and $H = 3$ ms.

Fig. 17 Effect of the step size to the solution accuracy; double-actuator model

- The interface model (IM) can allow a significant increase in the communication step size, but with a cost of degraded velocity level solution in some cases. The acceptability of this trade-off depends on the application, and therefore no definitive guidelines can be given regarding its use.
- If the step-size of the hydraulics integrator is small enough, here 0.1 ms, then the selection of the internal integrator seems to have no practical effect on the achievable communication step-size. In the examples reported in this research the multibody algorithm, the choice between polynomial extrapolation or use of an IM, and the selection of the coupling variables had a more noticeable effect on the achievable H when h_h was kept as the above-mentioned constant.
- Should the internal step-size of the hydraulic subsystem be increased, in turn, the results suggest that a relation exists between the largest achievable communication step-size and the internal step-size used in the fast subsystem. Thus, depending on the location of the computational bottleneck of the co-simulation setup, it may be possible to optimize the overall efficiency by tuning step-sizes in fast and slow subsystems together.

Conflict of Interest The authors declare that they have no conflict of interest.

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

References

1. Dopico, D., Luaces, A., Gonzalez, M., Cuadrado, J.: Dealing with multiple contacts in a human-in-the-loop application. Multibody Syst. Dyn. **25**(2), 167 (2011). https://doi.org/10.1007/s11044-010-9230-y
2. Baharudin, M.E., Rouvinen, A., Korkealaakso, P., Mikkola, A.: Real-time multibody application for tree harvester truck simulator. Proc. Inst. Mech. Eng., Part K: J. Multi-Body Dyn. **228**(2), 182 (2014). https://doi.org/10.1177/1464419314521182
3. Samin, J.C., Brüls, O., Collard, J.F., Sass, L., Fisette, P.: Multiphysics modeling and optimization of mechatronic multibody systems. Multibody Syst. Dyn. **18**(3), 345 (2007). https://doi.org/10.1007/s11044-007-9076-0
4. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: a survey. ACM Comput. Surv. **51**(3), 49:1 (2018). https://doi.org/10.1145/3179993.
5. Lacoursière, C., Härdin, T.: In: Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, pp. 653–662 (2017). https://doi.org/10.3384/ecp17132653
6. Sadjina, S., Kyllingstad, L.T., Rindarøy, M., Skjong, S., Æsøy, V., Pedersen, E.: Distributed co-simulation of maritime systems and operations. J. Offshore Mech. Arct. Eng. **141**(1), 011302 (2019). https://doi.org/10.1115/1.4040473
7. Benedikt, M., Hofer, A.: In: 2013 8th EUROSIM Congress on Modelling and Simulation, pp. 244–249 (2013). https://doi.org/10.1109/EUROSIM.2013.52
8. Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. Multibody Syst. Dyn. **4**(2), 107 (2000). https://doi.org/10.1023/A:1009810318420
9. 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 (2016). https://doi.org/10.1002/zamm.201400087
10. 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 (2018). https://doi.org/10.1007/s11044-018-9632-9
11. Oberschelp, O., Vöcking, H.: In: Proceedings of the IEEE International Conference on Mechatronics 2004, Istanbul, Turkey, pp. 404–409 (2004). https://doi.org/10.1109/ICMECH.2004.1364473
12. González, F., Naya, M.A., Luaces, A., González, M.: On the effect of multi-rate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. Multibody Syst. Dyn. **25**(4), 461 (2011). https://doi.org/10.1007/s11044-010-9234-7
13. Ben Khaled-El Feki, A., Duval, L., Faure, C., Simon, D., Gaid, M.B.: CHOPtrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. Simulation **93**(3), 185 (2017). https://doi.org/10.1177/0037549716684026
14. Rahikainen, J., González, F., Naya, M.A.: An automated methodology to select functional co-simulation configurations. Multibody Syst. Dyn. **48**(1), 79 (2020). https://doi.org/10.1007/s11044-019-09696-y

15. Sadjina, S., Kyllingstad, L.T., Skjong, S., Pedersen, E.: Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation. Eng. Comput. **33**(3), 607 (2017). https://doi.org/10.1007/s00366-016-0492-8

16. Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: In: NAFEMS World Congress 2013, Salzburg, Austria (2013)

17. Benedikt, M., Watzenig, D., Hofer, A.: Modelling and analysis of the non-iterative coupling process for co-simulation. Math. Comput. Model. Dyn. Syst. **19**(5), 451 (2013). https://doi.org/10.1080/13873954.2013.784340

18. 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). https://doi.org/10.1115/1.4028503

19. González, F., Arbatani, S., Mohtat, A., Kövecses, J.: Energy-leak monitoring and correction to enhance stability in the co-simulation of mechanical systems. Mech. Mach. Theory **131**, 172 (2019). https://doi.org/10.1016/j.mechmachtheory.2018.09.007

20. Haid, T., Stettinger, G., Watzenig, D., Benedikt, M.: In: Proceedings of the 5th Joint International Conference on Multibody System Dynamics, Lisbon, Portugal (2018)

21. Antunes, P., Magalhães, H., Ambrósio, J., Pombo, J., Costa, J.: A co-simulation approach to the wheel–rail contact with flexible railway track. Multibody Syst. Dyn. **45**(2), 245 (2019). https://doi.org/10.1007/s11044-018-09646-0

22. Antunes, P., Ambrósio, J., Pombo, J., Facchinetti, A.: A new methodology to study the pantograph-catenary dynamics in curved railway tracks. Veh. Syst. Dyn. Online first (2019). https://doi.org/10.1080/00423114.2019.1583348

23. Michaud, F., Lugrís, U., Cuadrado, J.: In: Proceedings of the 7 Congresso Nacional de Biomecânica, Guimarães, Portugal (2017)

24. Peiret, A., González, F., Kövecses, J., Teichmann, M.: Multibody system dynamics interface modelling for stable multirate co-simulation of multiphysics systems. Mech. Mach. Theory **127**, 52 (2018). https://doi.org/10.1016/j.mechmachtheory.2018.04.016

25. Naya, M.A., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and co-integration approaches. Arch. Mech. Eng. **58**(2), 223 (2011). https://doi.org/10.2478/v10180-011-0016-4

26. Rahikainen, J., Mikkola, A., Sopanen, J., Gerstmayr, J.: Combined semi-recursive formulation and lumped fluid method for monolithic simulation of multibody and hydraulic dynamics. Multibody Syst. Dyn. **44**(3), 293 (2018). https://doi.org/10.1007/s11044-018-9631-x

27. Rahikainen, J., Kiani, M., Sopanen, J., Jalali, P., Mikkola, A.: Computationally efficient approach for simulation of multibody and hydraulic dynamics. Mech. Mach. Theory **130**, 435 (2018). https://doi.org/10.1016/j.mechmachtheory.2018.08.023

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

29. Pan, D., Gu, S., Guo, G., Kuang, H., Zhong, H., Gao, F.: Co-simulation design and experimental study on the hydraulic–pneumatic-powered driving system of main steam and feed water isolation valves for CAP1400. Adv. Mech. Eng. **9**(8), 1 (2017). https://doi.org/10.1177/1687814017720078

30. Schmoll, R., Schweizer, B.: In: Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics, Brussels, Belgium (2011)

31. Bayo, E., García de Jalón, J., Serna, M.A.: A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems. Comput. Methods Appl. Mech. Eng. **71**(2), 183 (1988). https://doi.org/10.1016/0045-7825(88)90085-0

32. Bayo, E., García de Jalón, J., Avello, A., Cuadrado, J.: An efficient computational method for real time multibody dynamic simulation in fully Cartesian coordinates. Comput. Methods Appl. Mech. Eng. **92**, 377 (1991). https://doi.org/10.1016/0045-7825(91)90023-Y

33. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Comput. Methods Appl. Mech. Eng. **1**(1), 1 (1972). https://doi.org/10.1016/0045-7825(72)90018-7

34. González, F., Dopico, D., Pastorino, R., Cuadrado, J.: Behaviour of augmented Lagrangian and Hamiltonian methods for multibody dynamics in the proximity of singular configurations. Nonlinear Dyn. **85**(3), 1491 (2016). https://doi.org/10.1007/s11071-016-2774-5

35. Cuadrado, J., Cardenal, J., Morer Bayo, P.: Intelligent simulation of multibody dynamics: space–state and descriptor methods in sequential and parallel computing environments. Multibody Syst. Dyn. **4**(1), 55 (2000). https://doi.org/10.1023/A:1009824327480

36. Dopico, D., González, F., Cuadrado, J., Kövecses, J.: Determination of holonomic and nonholonomic constraint reactions in an index-3 augmented Lagrangian formulation with velocity and acceleration projections. J. Comput. Nonlinear Dyn. **9**(4), 041006 (2014). https://doi.org/10.1115/1.4027671

37. Watton, J.: Fluid Power Systems: Modeling, Simulation, Analog and Microcomputer Control. Prentice-Hall, New York (1989)
38. Handroos, H.M., Vilenius, M.J.: Flexible semi-empirical models for hydraulic flow control valves. J. Mech. Transm. Autom. Des. **113**(3), 232 (1991). https://doi.org/10.1115/1.2912774
39. Arnold, M., Clauß, C., Schierz, T.: In: Schöps, S., Bartel, A., Günther, M., ter Maten, E.J.W., Müller, P.C. (eds.) Progress in Differential-Algebraic Equations, pp. 107–125. Springer, Berlin (2014). https://doi.org/10.1007/978-3-662-44926-4_6