Multibody System Dynamics (2022) 55:103–136
https://doi.org/10.1007/s11044-022-09812-5

Energy-based monitoring and correction to enhance the accuracy and stability of explicit co-simulation

Borja Rodríguez[1] · Antonio J. Rodríguez[1] · Bernhard Sputh[2] · Roland Pastorino[2] · Miguel Ángel Naya[1] · Francisco González[1]

Received: 15 September 2021 / Accepted: 11 January 2022 / Published online: 27 January 2022

Abstract

The simulation of complex engineering applications often requires the consideration of component-level dynamics whose nature and time-scale differ across the elements of which the system is composed. Co-simulation offers an effective approach to deal with the modelling and numerical integration of such assemblies by assigning adequate description and solution methods to each component. Explicit co-simulation, in particular, is frequently used when efficient code execution is a requirement, for instance in real-time setups. Using explicit schemes, however, can lead to the introduction of energy artifacts at the discrete-time interface between subsystems. The resulting energy errors deteriorate the accuracy of the co-simulation results and may in some cases develop into the instability of the numerical integration process. This paper discusses the factors that influence the severity of the energy errors generated at the interface in explicit co-simulation applications, and presents a monitoring and correction methodology to detect and remove them. The method uses only the information carried by the variables exchanged between the subsystems and the co-simulation manager. The performance of this energy-correction technique was evaluated in multi-rate co-simulation of mechanical and multiphysics benchmark examples.

✉ B. Rodríguez
borja.rfrade@udc.es

A.J. Rodríguez
antonio.rodriguez.gonzalez@udc.es

B. Sputh
bernhard.sputh@siemens.com

R. Pastorino
roland.pastorino@siemens.com

M.Á. Naya
miguel.naya@udc.es

F. González
f.gonzalez@udc.es

[1] Laboratorio de Ingeniería Mecánica, University of A Coruña, 15403 Ferrol, Spain

[2] Model-Based System Testing, Siemens Digital Industries Software, Interleuvenlaan 68, 3001 Leuven, Belgium

Keywords Co-simulation · Energy monitoring · Energy correction · Energy-based indicators · Adaptive damping · Explicit Jacobi schemes

1 Introduction

Simulation has proven to be a valuable tool for the design, development and operation of engineering applications. Recent improvements in computing power and software tools have expanded its capabilities considerably, but expectations for its efficiency, the precision of the results that it delivers, and the complexity of the systems that it can successfully handle continue to grow as well. In the case of complex engineering applications such as automobiles, system-level simulation now needs to consider not only mechanical phenomena like suspension dynamics, but also processes of a different nature, such as hydraulics [5] or electronics [29], as well as the interactions between them. In many cases, it is possible to carry out the numerical integration of the dynamics of all these subsystems with a single, all-encompassing solver, e.g., [32], following a *monolithic* approach. Co-simulation [12], or solver coupling, is an alternative solution that features a number of advantages.

In a co-simulation setup, the overall system is divided into *subsystems* and a dedicated simulation tool, or *solver*, is assigned to each of them. Each solver takes care of the numerical integration of its subsystem, which enables the use of tailored modelling and solution methods. It is also possible to use different computing platforms, both in terms of hardware and software, to run each integration process [18]. The exchange of information between subsystems is limited to discrete instants in time known as *communication points*; the interval between two consecutive communication points is a *macro-step*. Often, only a reduced set of *coupling variables*, i.e., inputs and outputs, is transferred at these instants. Between the communication points, the integration of the dynamics of each subsystem proceeds without additional data from its environment. The information about the subsystem internals is thus not disclosed to third-party software, which is an attractive feature from the point of view of preserving intellectual property rights in collaborative projects. In this regard, the seamless coupling of software tools from different vendors has been simplified by the introduction of interface standards like FMI [1]; in fact, in many industrial applications co-simulation subsystems can effectively be considered to behave as *black boxes*. Moreover, certain co-simulation schemes allow the distribution of the computational workload between several processing units, which can be used to parallelize code execution and improve efficiency [20].

Co-simulation setups can be built according to a wide variety of schemes and configuration options; the computational performance and accuracy of the results are considerably affected by this selection [24, 27]. Two major groups of coupling schedules can be identified: implicit and explicit ones. In implicit co-simulation schemes, macro-steps are repeated to improve the convergence of results. Explicit methods, conversely, do not perform rollback and macro-steps cannot be retaken once their integration has finalized. Implicit co-simulation methods are typically more stable and accurate than their explicit counterparts [17, 33]; nonetheless, noniterative algorithms are the option of choice in some cases. Efficient execution may be incompatible with iteration over macro-steps, for instance in applications in which the available time for computations is limited. In other cases, one or more subsystems do not allow rollback. Cyber-physical systems, in which physical components interact with virtual environments, are an example of an application subjected to these two constraints. On the one hand, all the subsystems in the setup must deliver real-time performance; depending on the complexity of the computational models used, some virtual components may not be able to meet this requirement if they have to iterate over their macro-steps. Besides, it is generally not possible to revert the state of a physical component to a previous instant in time.

Discrete-time communication at the co-simulation interface has been identified as a source of discontinuities [10] and delays [7] in the inputs that the subsystems receive at the start of their macro-steps. These issues cause the co-simulation results to deviate from theoretically correct solutions [19], which translates into violations of the energy balance of the system [14], particularly when explicit schemes are used. This can make the simulation results unreliable and, in extreme cases, render the numerical integration unstable. Techniques like input extrapolation [23] are commonly used to alleviate this problem. Subsystem inputs can also be evaluated employing physics-based predictions delivered by reduced subsystem models [25, 26], or approximated using directional derivatives that relate coupling variables and subsystem states [15, 16, 35]. Input prediction can deliver significant improvements with regard to co-simulation quality. It is, however, difficult to remove coupling errors completely, or even quantify them using this technique alone with explicit schemes.

Input correction is an alternative means of preventing the degradation of co-simulation results, based on modifying the subsystem inputs so that they comply with certain physics-based criteria. This is the strategy followed in [8], which introduces a compensation signal to remove energy inconsistencies derived from input extrapolation. Another correction method was presented in [13], that enforces the conservation of the energy balance of the whole system by means of passivity-based control, assuming that information about the internal energy state of each subsystem is available. Control criteria can also be used to define similar correction algorithms [11]. An advantage of this strategy is that corrections are performed based on information about the system that can be used to evaluate co-simulation quality. Energy-based indicators can be used to gather this information: The concepts of power residual and energy residual were introduced in [30]. They were initially used to define algorithms to control the macro-step size, but can also serve as a basis to develop methods to correct the system energy [31].

This paper puts forward an energy-monitoring and input-correction method for explicit co-simulation that uses the information conveyed by the coupling variables exchanged between the subsystems. This algorithm can be used with Jacobi schemes in which the product of input and output variables has power units, e.g., in force-displacement coupling applications. The ability of the proposed approach to keep the co-simulation stable and accurate was evaluated by means of a set of benchmark problems, which included mechanical and multiphysics systems, both linear and nonlinear.

Sect. 2 in this paper provides an overview of the features and issues of explicit Jacobi co-simulation schemes. A discussion of the problems with energy conservation derived from the use of discrete co-simulation interfaces in these cases, together with the indicators used to monitor them, can be found in Sect. 3. A correction method for removing energy artifacts is presented here as well. Section 4 shows the application of the indicators and the energy correction algorithm to several explicit co-simulation scenarios. The results are discussed in Sect. 5; Sect. 6 summarizes the conclusions of the research.

2 Explicit Jacobi co-simulation schemes

Jacobi schemes are commonly used in explicit co-simulation. This arrangement allows for simultaneous integration of subsystems, a convenient feature for the parallelization of code execution, which becomes more complicated if sequential Gauss-Seidel schemes are adopted. Figure 1 illustrates the way in which the numerical integration proceeds between

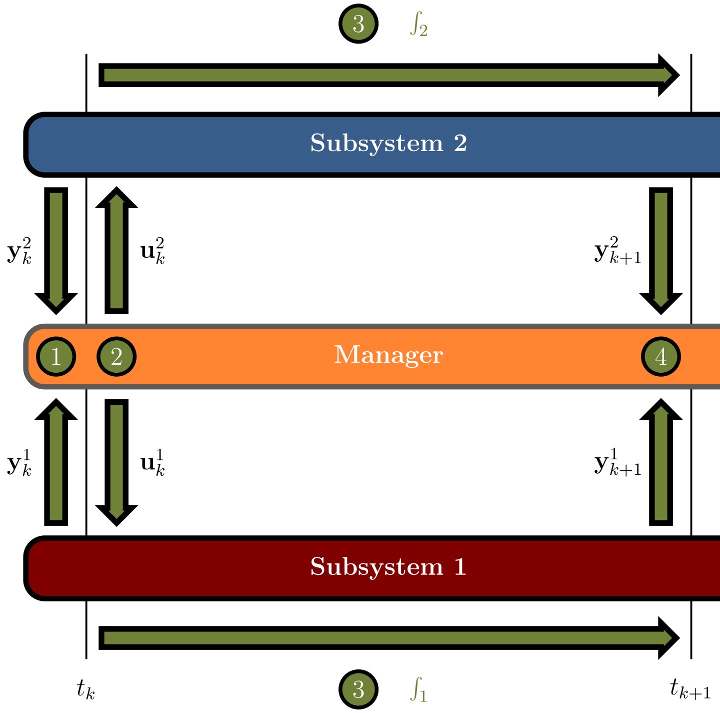

Fig. 1 Diagram of an explicit, single-rate Jacobi coupling scheme with two subsystems

two communication points according to this scheme, for an application with two subsystems. At time $t = t_k$, the subsystems submit their outputs $\mathbf{y}_k^1$ and $\mathbf{y}_k^2$ to the co-simulation manager (step 1), which in turn evaluates and sends back the inputs $\mathbf{u}_k^1$ and $\mathbf{u}_k^2$ to them (step 2). At this point, both subsystems can independently carry out their integration from time t_k to t_{k+1} (step 3). The process is restarted at time t_{k+1} when the new outputs $\mathbf{y}_{k+1}^1$ and $\mathbf{y}_{k+1}^2$ are forwarded to the manager (step 4).

Jacobi schemes can be single-rate if the communication step-size is the same for all subsystems, or multi-rate if different step-sizes are used instead. The energy correction presented in this paper is evaluated differently in each case; direct feedthrough is another factor that needs to be taken into consideration as well.

The concept of direct feedthrough denotes the explicit dependence of the subsystem output from its input. Its presence in a co-simulation scheme may lead to the existence of algebraic loops, which often motivates the necessity to perform the initialization stage in an iterative way [2]. Besides, during runtime, direct feedthrough is associated with a deterioration of the convergence and stability properties of numerical integration [4].

Direct feedthrough motivates the need to extrapolate subsystem inputs in explicit single-rate Jacobi schemes, even when single step integrator formulas are used inside the subsystems. This extrapolation often becomes a source of coupling errors. The issue can be illustrated assuming that the dynamics of a given subsystem can be expressed using the discrete state-space notation as follows

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \quad (1)$$

where $\mathbf{x}$ is the subsystem state, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are the state, input, output and feedthrough matrices, respectively. In a single-rate Jacobi scheme like the one in Fig. 1, if explicit integration formulas are used in the subsystems, the evaluation of the state derivatives $\dot{\mathbf{x}}$ in the macro-step $[t_k, t_{k+1}]$ only needs to be performed at instant t_k to evaluate $\mathbf{x}_{k+1}$. At this time, both the state $\mathbf{x}_k$ and the input $\mathbf{u}_k$ are known, and so the discrete-time interface does not introduce any error in the computation of the system derivatives.

The subsystem output $\mathbf{y}$, on the other hand, must be evaluated and returned at time t_{k+1} as

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} + \mathbf{D}\mathbf{u}_{k+1} \tag{2}$$

and this in principle requires input $\mathbf{u}_{k+1}$, which is not available yet. If the subsystem has no direct feedthrough, i.e., $\mathbf{D} = \mathbf{0}$, the output $\mathbf{y}_{k+1}$ can be directly evaluated as

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} \tag{3}$$

The outputs delivered by Eq. (3) only include the numerical error caused by the integration formula used to update the state between t_k and t_{k+1}. On the contrary, a subsystem with direct feedthrough will feature $\mathbf{D} \neq \mathbf{0}$ and an approximation $\tilde{\mathbf{u}}_{k+1}$ of the inputs at t_{k+1} is necessary during output evaluation,

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} + \mathbf{D}\tilde{\mathbf{u}}_{k+1} \tag{4}$$

The approximated input $\tilde{\mathbf{u}}_{k+1}$ can be obtained in different ways, e.g., via polynomial extrapolation, but will generally not equal $\mathbf{u}_{k+1}$. This approximation introduces an additional error in the evaluation of the system dynamics, which deteriorates the accuracy of the co-simulation results.

However, there exist several scenarios in which input extrapolation introduces additional sources of error into the co-simulation process, even if the subsystems are not subjected to direct feedthrough:

- Implicit integration formulas require the derivative $\dot{\mathbf{x}}_{k+1}$ in the evaluation of $\mathbf{x}_{k+1}$. Input approximation is necessary for this, as evidenced by Eq. (1).
- Multi-rate co-simulation schemes make it necessary to evaluate the derivatives of the subsystem state at intermediate points between the communication points t_k and t_{k+1}, giving rise to the need for input extrapolation as well, except in the rather infrequent case in which $\mathbf{B} = \mathbf{0}$.
- In the case of nonlinear systems, matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ may be functions of the state $\mathbf{x}$ and the input $\mathbf{u}$.

The above-mentioned circumstances have to be taken into account for a correct evaluation of the energy imbalance at the coupling interface, besides other errors, e.g., those that stem from the integration process, which also introduce inaccuracies in the results, although their impact is frequently less critical than that of input extrapolation [11]. These issues will be addressed in Sect. 3, which discusses how to calculate the energy errors introduced by co-simulation and puts forward a method to eliminate this artificial modification of the system energy.

3 Energy-based indicators for explicit co-simulation

Determining the stability of generic co-simulation applications is a complicated task. Quantifying how much they deviate from theoretically correct results is even more challenging.

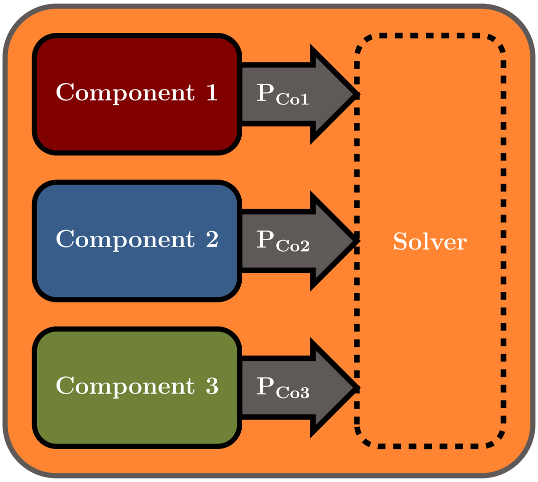

Fig. 2 Power flows within a three-component monolithic solver

Because such reference solutions are often not available for comparison, indicators have been proposed and employed to convey this information.

If the integration formulas used inside the subsystems are known, it is possible to evaluate the stability of the coupling scheme by its spectral radius [10, 33, 36]. A measure of stability, however, does not include all relevant information about the accuracy of the results. The convergence of particular coupling methods and schemes can be evaluated, e.g., [4, 35], but it is not straightforward to use these to quantify the actual error introduced by a particular simulation case, although error bounds can be defined in particular cases [3]. Some methods to quantify the errors introduced in the co-simulation results have been put forward nonetheless. In general, these require some information about the subsystem internals, either directly obtained [20] or provided by the directional derivatives of the subsystems [15].

In many explicit co-simulation applications, the subsystems behave as black-boxes and so the information about their internals is not accessible to the manager. In these cases, it is useful to introduce indicators that can be evaluated using only the information carried by the coupling variables. This is the case of the residual power and energy introduced in [30], defined for co-simulated systems in which the product of input and output variables has units of power. A method to adjust the macro step-size of the setup was developed based on these, which was later [31] combined with an energy removal method similar to the one in [8] to improve the accuracy of the results. A similar energy-correction algorithm can be found in [13]; this one, however, requires information about the energy balance of the overall co-simulated system.

The power and energy residuals are used next as indicators for co-simulation quality and an interpretation of their physical meaning is provided. Then, an energy-correction method is developed and its performance is shown with a set of co-simulation benchmark problems.

3.1 Energy-based indicators for co-simulation

The schematic of a monolithic solver for a system with three components is depicted in Fig. 2.

If the numerical integration errors are neglected, the overall energy balance for a mechanical system at any instant in time t is given by

$$E(t) - E_0 - W_{\mathrm{nc}}(t) = 0 \tag{5}$$

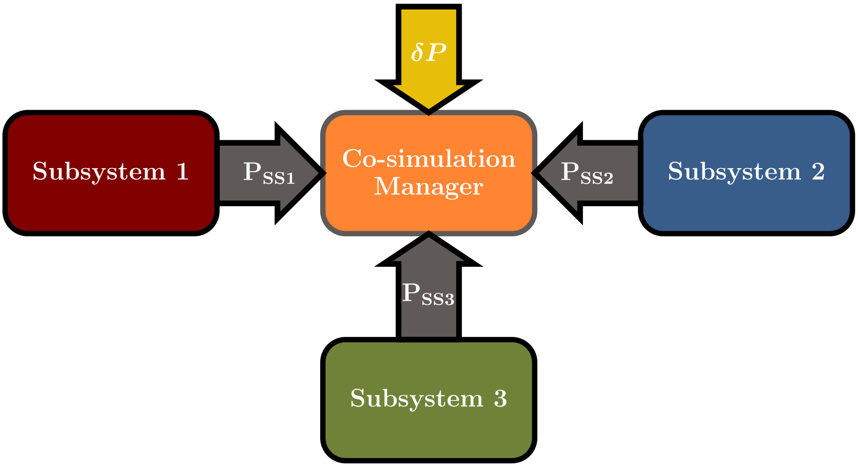

Fig. 3 Power balance of a three-subsystem co-simulation environment

where E is the mechanical energy, E_0 is the initial energy level, and W_{nc} is the work of nonconservative forces. Ideally, Eq. (5) must hold at all times. Thus, the power exchange between the three components in Fig. 2 must satisfy the conservation of energy, which is expressed as

$$\sum_{i=1}^{n_{\mathrm{co}}} P_{Coi} = P_{Co1} + P_{Co2} + \cdots + P_{Con_{\mathrm{co}}} = 0 \tag{6}$$

where P_{Coi}, $i = 1, 2, \ldots, n_{\mathrm{co}}$, is the power exchanged between each component and the rest of the system, and n_{co} is the number of components in the system. Equation (6) means that, in the absence of external sources or sinks of energy, all the power flows without losses at the interface between the components. This monolithic scheme contains a single integrator, which takes care of the complete system and is allowed to access the internal information, e.g., the state, of each component at any point in time.

If a co-simulation scheme is used instead, as illustrated in Fig. 3, the power exchanged at the interface should also verify Eq. (6). Again, even if numerical integration errors are neglected, the system energy is modified by the co-simulation scheme due to the discrete-time communication between the manager and the subsystems [13, 30]. The violation of the power conservation can be expressed by means of the *residual power* δP as

$$\sum_{i=1}^{n_{\mathrm{ss}}} P_{SSi} = P_{SS1} + P_{SS2} + \cdots + P_{SSn_{\mathrm{ss}}} = -\delta P \neq 0 \tag{7}$$

where P_{SSi}, $i = 1, 2, \ldots, n_{\mathrm{ss}}$, is the power exchanged between a subsystem and the co-simulation manager through the interface, and n_{ss} is the number of subsystems.

The residual power δP can be used to assess co-simulation quality when the product of subsystem inputs and outputs has units of power. Its evaluation does not require access to the subsystem internals, but only the information carried by the coupling variables. This is the case for a wide range of practical co-simulation setups, for instance, force-displacement or torque-angular speed couplings in mechanical systems or voltage-current exchanges in electric or electronic circuits.

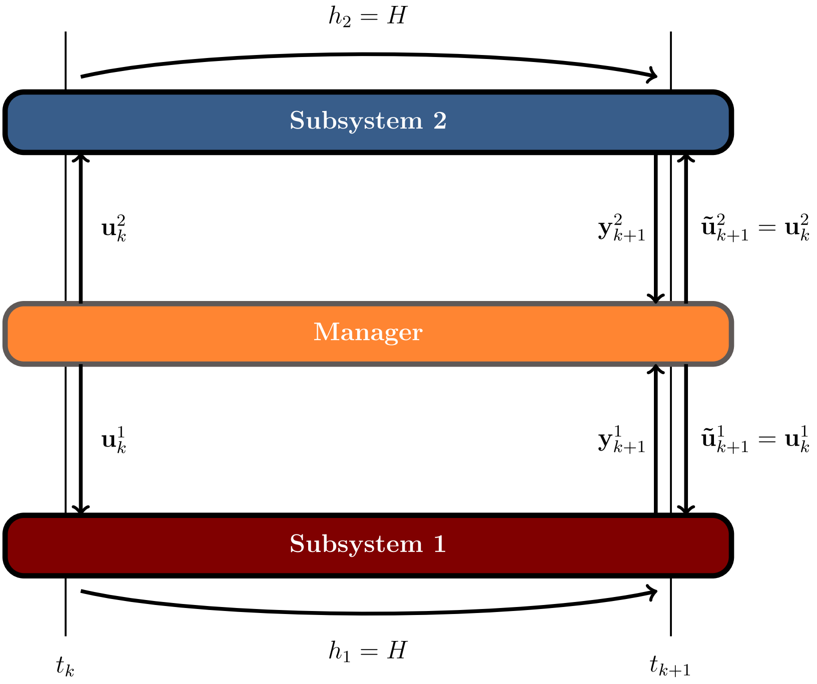

Fig. 4 Evaluation of the residual power δP in a single-rate, explicit Jacobi scheme

For some selections of coupling variables, the input-output product could also deliver other units related to power exchange, e.g., $\mathrm{W/m^2}$ s. In these situations, it would also be possible to evaluate the residual power, with the additional requirement that the co-simulation manager would need to know some physical properties of the subsystems.

Assuming that the product of the coupling variables in a certain co-simulation configuration has power units, it is possible to evaluate the residual power δP for single-rate Jacobi schemes with two subsystems as [30]

$$\delta P_{k+1} = -\left(\mathbf{u}_{k+1}\right)^{\mathrm{T}} \mathbf{y}_{k+1} \tag{8}$$

where $\mathbf{u}$ and $\mathbf{y}$ are the arrays of inputs and outputs handled by the co-simulation manager. Ideally, Eq. (8) should equal zero if no errors existed at the coupling interface.

Figure 4 shows power exchanges in a single-rate explicit Jacobi schedule. The superscripts $()^1$ and $()^2$ denote the first and second subsystems, whereas subscripts $()_k$ and $()_{k+1}$ refer to the time-step indexes t_k and t_{k+1}, respectively. This figure also illustrates the issue with the evaluation of subsystem outputs at time t_{k+1}. As already discussed in Sect. 2, the input $\mathbf{u}_{k+1}$ is necessary to determine outputs $\mathbf{y}_{k+1}$ in subsystems with direct feedthrough. In fact, $\mathbf{u}_{k+1}$ is also needed to complete the integration of the dynamics from t_k to t_{k+1} if implicit integrators like the Newmark family [22] are employed. For simplicity, the discussion here is limited to subsystems with explicit integrators, e.g., the symplectic Euler formula.

Several ways to approximate $\mathbf{u}_{k+1}$ can be used; polynomial extrapolation is widely used for this purpose [9, 23, 27], especially low-order methods such as constant or zero-order hold (ZOH), linear or first-order hold (FOH), and quadratic or second-order hold (SOH), as

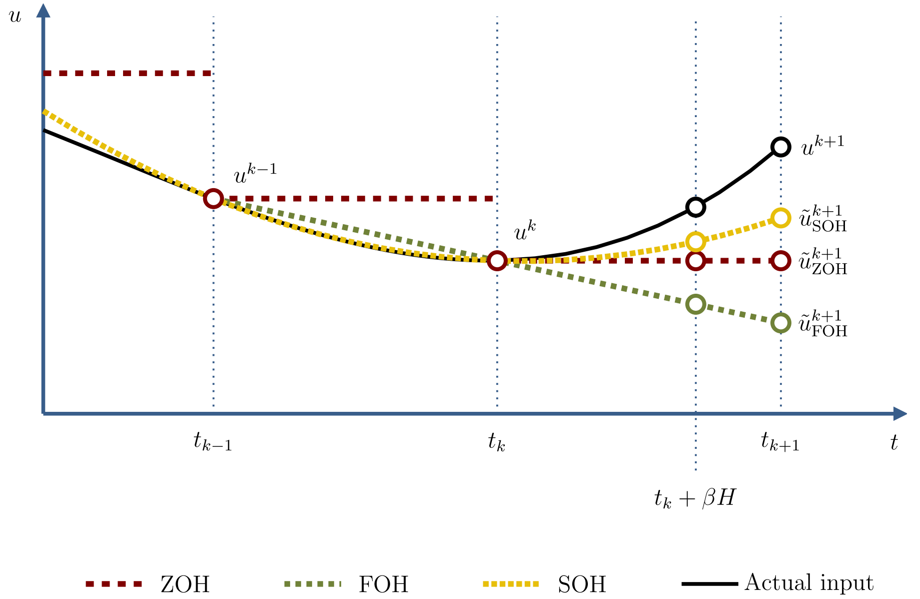

Fig. 5 Effect of ZOH, FOH and SOH extrapolation on the prediction of a subsystem input u between t_k and t_{k+1}, where $\beta \in [0, 1]$

shown in Fig. 5. Accordingly, outputs $\mathbf{y}_{k+1}$ are evaluated with an approximation $\tilde{\mathbf{u}}_{k+1}$ instead of the input value $\mathbf{u}_{k+1}$ and the residual power becomes

$$\delta P_{k+1} = -\left(\tilde{\mathbf{u}}_{k+1}\right)^{\mathrm{T}} \mathbf{y}_{k+1} \tag{9}$$

which, in general, will no longer satisfy $\delta P = 0$.

The evaluation of residual power corresponds to the co-simulation manager; in general, it cannot be assumed that all subsystems evaluate their residual power and include it among their coupling variables. This means that some assumption has to be made about the method that each subsystem uses to approximate its inputs $\mathbf{u}_{k+1}$. Here we consider that $\tilde{\mathbf{u}}_{k+1} = \mathbf{u}_k$ in the evaluation of δP in Eq. (9).

The definition of the indicator in Eq. (9) can be generalized to Jacobi multi-rate coupling schemes as well [30].

The residual power δP can be used to keep track of the stability and accuracy of the integration process as the simulation proceeds. The indicator values can, for example, oscillate within an admissible range, which will be dependent on the application. In such cases, the co-simulated dynamics can be considered reasonably accurate. Conversely, the presence of sudden changes in the indicator can be a symptom of unstable behaviour and degraded accuracy of the results.

Moreover, the residual power δP is related to the accumulation of errors in the energy balance of the system. The deviations from the exact solution caused by input extrapolation will introduce an error in the balance in Eq. (5),

$$E(t) - E_0 - W_{\mathrm{nc}}(t) = \varsigma(t) \tag{10}$$

where $\varsigma(t)$ is the energy deviation at time t caused by the discrete-time interface.

The existence of a relation between the energy error ς and the residual power δP was confirmed in [27]. The error ς, however, cannot be directly obtained by integrating the δP over time. As a matter of fact, depending on the selected coupling scheme and the properties of the subsystems, it is possible to approximate it with a fraction $\mu \in [0, 1]$ of this integral. For example, for the first co-simulation step, the violation of the energy balance is given by

$$\varsigma\left(t_1\right) \approx \mu \int_{t_0}^{t_1} \delta P \, \mathrm{d}t \approx \mu \, H \, \delta P_1 \approx -\mu \, H \, \mathbf{u}_0^{\mathrm{T}} \, \mathbf{y}_1 \tag{11}$$

As integration proceeds over time, however, the relation given by μ may lose accuracy because the accumulation and propagation of errors brings the numerical integration further away from the theoretical solution. For this reason, the integral over time of the residual power, i.e., the residual energy $\delta E\left(t\right)$, does not directly provide the value of the energy error $\varsigma\left(t\right)$ in the system,

$$\delta E\left(t\right) = \int_{t_0}^{t} \delta P\left(t\right) \, \mathrm{d}t \neq \varsigma\left(t\right) \tag{12}$$

and the use of expression (11) must be limited to the calculation of the energy error introduced by the co-simulation environment between two consecutive communication points. A method for determining the value of μ is discussed in Sect. 3.3.

3.2 Energy correction

The information conveyed by the residual power δP can be used to develop an energy-correction method aimed at removing the energy deviations derived from input extrapolation. The presented method is similar to the energy correction solution described in [13], in the sense that its objective is enforcing the satisfaction of the energy balance of the complete system, dissipating excess energy or introducing it back into the system if it has been lost, as an adaptive damping element would do. A major difference is that this method does not require the subsystems to provide information about their internal energy state, which is not possible in a number of practical applications. The co-simulation macro step-size is not modified.

As mentioned in Sect. 3.1, the integral of the residual power δP over time does not correspond exactly to the accumulated energy error ς of the overall system and needs to be corrected by means of a coefficient μ. The energy error between two consecutive communication points can then be approximated as

$$\varsigma_{k+1} \approx \mu \, \delta E_{k+1} \; = \mu \int_{t_k}^{t_{k+1}} \delta P\left(t\right) \mathrm{d}t \tag{13}$$

where ς_{k+1} denotes the energy error added to the system between communication points t_k and t_{k+1}, and δE_{k+1} is the residual energy evaluated at time t_{k+1}. Ideally, the energy error in Eq. (13) should be removed by the corrective action acting during the following macro step-size, from t_{k+1} to t_{k+2}, in order to readjust the energy level of the system. The nature of this action depends on the kinds of coupling variables being exchanged at the interface; in a force-displacement configuration, for instance, it is a force.

Figure 6 conceptually illustrates the correction scheme used. In this co-simulation graph, the output of subsystem 1, $\mathbf{f}$, is an *action*, e.g., a force or torque. The output of subsystem 2, $\mathbf{v}$, in turn, represents a kinematic quantity such as a velocity. It will be assumed in the

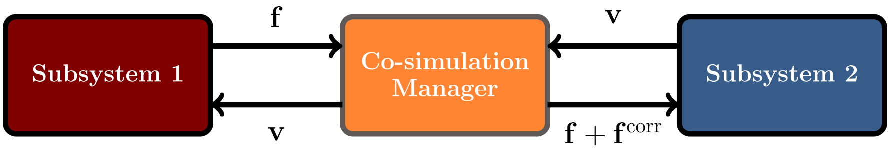

Fig. 6 Energy correction via modification of the coupling variables

following that the product $\mathbf{f}^{\mathrm{T}}\mathbf{v}$ has power units. In general, subsystems that return actions are more likely to present direct feedthrough, as they often need to know the position or velocity of other subsystems to evaluate their outputs. The proposed correction method modifies the action $\mathbf{f}$ sent to the second subsystem with an additional term $\mathbf{f}^{\text{corr}}$ that corrects the energy error from the previous macro step,

$$-\varsigma_{k+1} = \mathbf{f}^{\text{corr}}_{k+1}\,\mathbf{v}_{k+1}\,H \quad\rightarrow\quad \mathbf{f}^{\text{corr}}_{k+1} = -\frac{\varsigma_{k+1}}{\mathbf{v}_{k+1}\,H} \tag{14}$$

The action correction in Eq. (14) can be compared to the introduction of adaptive damping at the co-simulation interface. The applicability of Eq. (14) is limited by several facts. First, values of the coupling variable $\mathbf{v}$ that approach zero will result in large values of the correction term $\mathbf{f}^{\text{corr}}$, which can cause impulsive behaviour and destabilize the system dynamics. The maximum admissible correction must be limited in practical applications. As a consequence, the possibility of removing energy errors is limited at these points. It must be noted that the force limit used is problem-dependent and depends on the nature and time-scale of the subsystem dynamics. In this work it was set to a percentage of the uncorrected coupling force.

Additionally, the correction term $\mathbf{f}^{\text{corr}}$ is calculated using the values of energy and velocity obtained at the end of the macro step $[t_k, t_{k+1}]$, but the correction action $\mathbf{f}^{\text{corr}}_{k+1}$ has to be applied during the next macro step $[t_{k+1}, t_{k+2}]$, at which the value $\mathbf{v}_{k+1}$ used to evaluate the correction is unlikely to remain constant. For this reason, the energy correction resulting from the introduction of $\mathbf{f}^{\text{corr}}_{k+1}$ will not match exactly the value of ς_{k+1} that it intended to remove, but a difference U_{k+1} will exist between them. A possible way to approximate this value is to evaluate the difference between the intended correction and the one that would result if the coupling velocity at time t_{k+2} were used

$$U_{k+1} \approx \varsigma_{k+1} + \mathbf{f}^{\text{corr}}_{k+1}\,\mathbf{v}_{k+2}\,H \tag{15}$$

The term U_{k+1} in Eq. (15) can be seen as the energy deviation that the method has been unable to correct during the last macro step. The accumulation of these terms over time leads to an uncorrected energy deviation $\mathcal{U}$ in the overall system dynamics

$$\mathcal{U}_{k+1} = \mathcal{U}_k + U_{k+1} \quad ; \quad \text{where} \quad \mathcal{U}_0 = 0\,\mathrm{J} \tag{16}$$

An extra term can be introduced in the correction algorithm in Eq. (14), to remove the accumulated energy difference $\mathcal{U}_k$

$$\mathbf{f}^{\text{corr}}_{k+1} = -\frac{\varsigma_{k+1} + \nu\,\mathcal{U}_k}{\mathbf{v}_{k+1}\,H} \tag{17}$$

where ν is a weight coefficient similar to an integral gain in the range [0, 1], where 0 stands for no integral correction and 1 would enforce the removal, in theory, of all the accumulated energy error $\mathcal{U}_k$ in the next macro step.

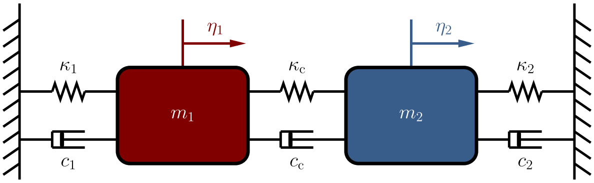

Fig. 7 A generic two-degree-of-freedom oscillator

Table 1 Combinations of system parameters used in double linear oscillator benchmarks

Case	m_1 [kg]	m_2 [kg]	κ_1 [N/m]	κ_c [N/m]	κ_2 [N/m]	c_1 [Ns/m]	c_c [Ns/m]	c_2 [Ns/m]
1	1	1	10	100	1000	0	0	0
2	1	1	10	100	1000	0.1	0.1	0.1
3	1	1	10	1000	100	0	0	0
4	1	1	10	1000	100	0.1	0.1	0.1
5	1	1	1000	100	10	0	0	0
6	1	1	1000	100	10	0.1	0.1	0.1
7	10	1	10	100	1000	0	0	0
8	1	10	10	100	1000	0	0	0

It is also worth mentioning that in co-simulation environments that include physical components, the introduction of a correction term $\mathbf{f}^{\mathrm{corr}}$ has to be compatible with the nature of actuators used in the system. Additional improvements could be performed on the method, such as using other integration formulas in Eq. (13) or conducting some prediction of the velocity during the next communication step when calculating the correction action in Eq. (14).

3.3 Evaluation of correction coefficients

The indicators and the correction method presented in Sects. 3.1 and 3.2 require the determination of the fraction μ of the residual power δP that indicates the spurious energy introduced into the system at the co-simulation interface.

The preliminary evaluation of the μ coefficient was performed with the help of a two-degree-of-freedom linear oscillator, shown in Fig. 7, a system widely used in the co-simulation literature as benchmark problem, e.g., [12–14, 33].

The double linear oscillator is a mechanical system composed of two masses (m_1 and m_2) connected to each other and to the ground by means of dampers (c_1, c_2 and c_c) and springs (κ_1, κ_2, and κ_c) with constant coefficients. The initial system displacements were set to $\eta_1(0) = \eta_2(0) = 0\,\mathrm{m}$; the spring forces are zero in this configuration. The initial system velocities were $\dot{\eta}_1(0) = 100\,\mathrm{m/s}$ and $\dot{\eta}_2(0) = -100\,\mathrm{m/s}$, a set of values previously used in the literature, e.g., [13, 34, 35]. Several simulation cases, summarized in Table 1, were defined to verify the system behaviour for different sets of physical parameters.

Figure 8 shows the linear oscillator split into two subsystems in a force-displacement co-simulation configuration. The subsystem $\mathcal{M}_1$ includes the coupling spring-damper system and returns the force $\mathbf{f}$ that it exerts on the second subsystem, $\mathcal{M}_2$. This, in turn, takes care of the integration of the motion of m_2 and delivers its displacement η_2 and velocity $\dot{\eta}_2$, denoted here as the array $\mathbf{x}_2$.

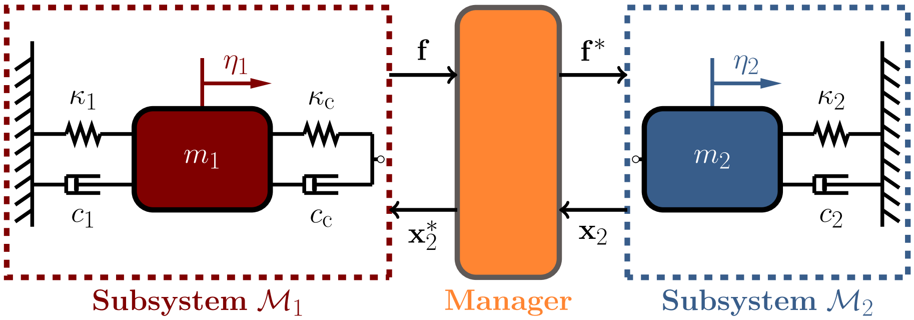

Fig. 8 The generic two-degree-of-freedom oscillator arranged following a force-displacement coupling scheme

Only the subsystem $\mathcal{M}_1$ is subjected to direct feedthrough ($\mathbf{D}_{\mathcal{M}_1} \neq \mathbf{0}$), and requires to know its inputs to evaluate its outputs at any given instant in time. In this scheme, the inputs $\mathbf{u}$ and outputs $\mathbf{y}$ for each subsystem are given by

$$\begin{aligned} \mathbf{u}_{\mathcal{M}_1} &= \left[\mathbf{x}_2^*\right] = \left[\eta_2^* \; \dot{\eta}_2^*\right]^{\mathrm{T}} & \qquad \mathbf{u}_{\mathcal{M}_2} &= \left[\mathbf{f}^*\right] = \left[f^*\right] \\ \mathbf{y}_{\mathcal{M}_1} &= \left[\mathbf{f}\right] = \left[f\right] & \qquad \mathbf{y}_{\mathcal{M}_2} &= \left[\mathbf{x}_2\right] = \left[\eta_2 \; \dot{\eta}_2\right]^{\mathrm{T}} \end{aligned} \tag{18}$$

where the superscript $()^*$ points out that the input of a subsystem might not necessarily be equal to the output of the other one, because the co-simulation manager may perform modifications in their values, e.g., by extrapolation.

The double linear oscillator can be used to illustrate the relation between the residual energy, δE, and the actual energy error ς accumulated during the co-simulation. Figures 9 and 10 compare the mechanical energy E of the double linear oscillator co-simulated following an explicit, single-rate Jacobi scheme with the mechanical energy obtained with its analytical solution, E^{ref}. The shown results correspond to cases 1 and 2 in Table 1.

These figures confirm that the residual power δP and its integral δE indicate the deviation of the co-simulated system dynamics from its theoretical energy balance. In conservative systems, this results into an increase in mechanical energy. In systems with dissipation, part of the energy error modifies the mechanical energy, and part affects the work exerted by nonconservative forces. Figure 10 highlights two interesting facts. First, the results show that the co-simulation is *stable*, in the sense that the system energy decreases with time; however, they also evidence that it is not *accurate*, because the system energy does not follow that of the reference solution. Second, the presence of dissipation within the subsystems decreases the severity of the energy errors introduced by the time-discrete coupling interface, bringing the co-simulated solution closer to its theoretically correct counterpart than in the conservative case.

The linear oscillator example was used to determine the value of the fraction μ required to satisfy Eq. (11) and, thus, achieve an accurate correction of the system energy by means of Eq. (14). This was carried out through the simulation of case 1 in Table 1. Different integration and communication step-sizes were used; in multi-rate co-simulation scenarios, ZOH, FOH and SOH input extrapolation orders were tested. The symplectic Euler formula was used as the integration method in all the tests. The values of μ obtained this way were used as a reference for the simulation of the nonlinear benchmark examples presented in Sect. 4 and the rest of cases described in Table 1. For the linear oscillator, this coefficient

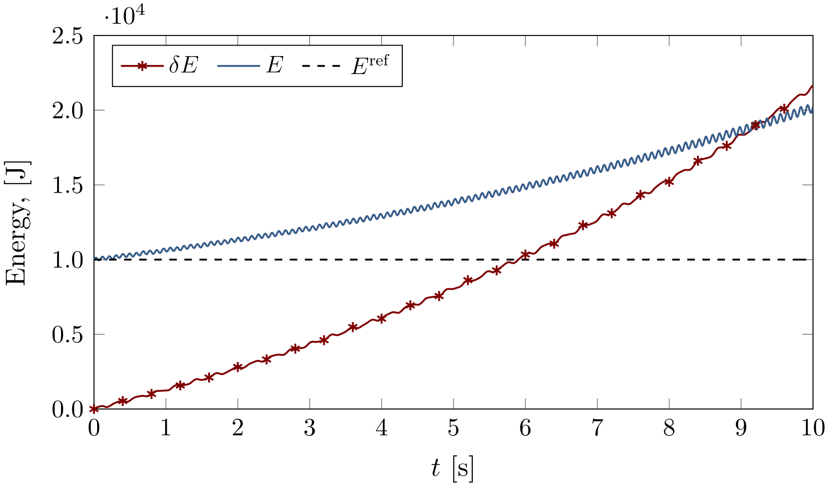

Fig. 9 Comparison of the residual energy δE, the co-simulated system energy E, and the reference solution energy E^{ref} in a force-displacement, single-rate Jacobi coupling scheme for a mechanical system without damping

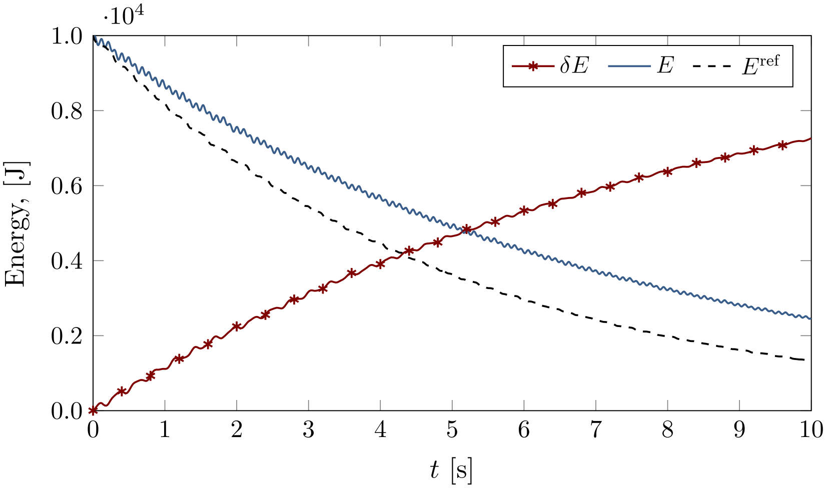

Fig. 10 Comparison of the residual energy δE, the co-simulated system energy E, and the reference solution energy E^{ref} in a force-displacement, single-rate Jacobi coupling scheme for a mechanical system that features damping

was found to depend on the existence of direct feedthrough in the subsystems. In single-rate schemes, the coefficient μ equals 0.5 if one of the subsystems is subjected to direct feedthrough [30]; its value is zero if both subsystems are free from this issue, because input extrapolation at the interface does not take place and energy correction is not necessary. On the other hand, in multi-rate co-simulation schemes, the ratio between the macro step-sizes and the extrapolation order at the co-simulation manager also needs to be considered. In these schemes, however, the evaluation of the coefficient μ becomes more complicated

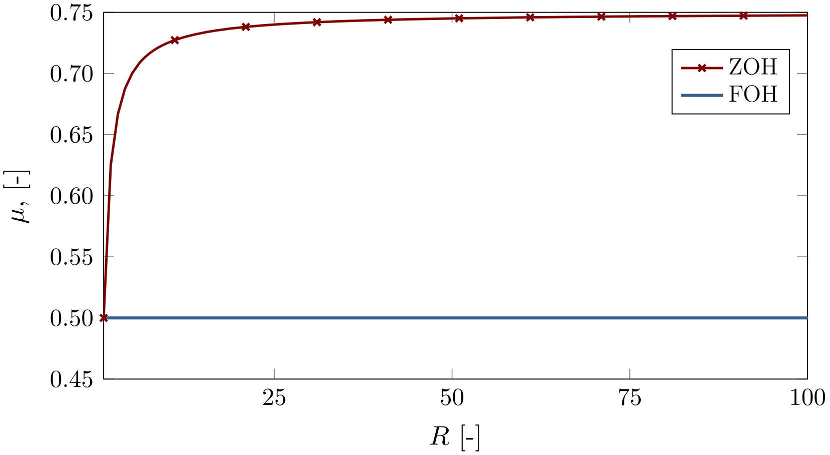

Fig. 11 Selection of the correction factor μ in a two-subsystem co-simulation, as a function of the step-size ratio and the extrapolation order when the subsystem with the largest step-size has direct feedthrough

and it is necessary to consider the extrapolation order and the ratio R between subsystem step-sizes

$$R = \frac{h_{\mathrm{df}}}{h_{\mathrm{ndf}}} \tag{19}$$

where h_{df} and h_{ndf} are the macro step-sizes of the subsystems with and without direct-feedthrough, respectively. The value of the coefficient μ can then be calculated using the following expressions for a two-subsystem co-simulation:

- *The subsystem with direct feedthrough has the largest step-size:* the corrective coefficient is obtained as

$$\begin{aligned} \mu_{\mathrm{ZOH}} &= \mu_{\mathrm{SR}}\,(1.5 - \frac{0.5}{R}) \\ \mu_{\mathrm{FOH}} &= \mu_{\mathrm{SR}} \end{aligned} \tag{20}$$

- *The subsystem with direct feedthrough has the smallest step-size:* the corrective coefficient is obtained as

$$\begin{aligned} \mu_{\mathrm{ZOH}} &= \mu_{\mathrm{SR}} \\ \mu_{\mathrm{FOH}} &= \mu_{\mathrm{SR}}\, R \end{aligned} \tag{21}$$

Subscripts ZOH (zero-order hold) and FOH (first-order hold) denote the input extrapolation order used in the subsystem with the smallest step-size.

Figures 11 and 12 show the values of μ calculated with Eqs. (20) and (21) for different values of the step-size ratio R and the two mentioned input extrapolation methods. Although it is not represented in the figure, the μ for higher order extrapolation, e.g., quadratic or SOH, is the same as the one used for FOH.

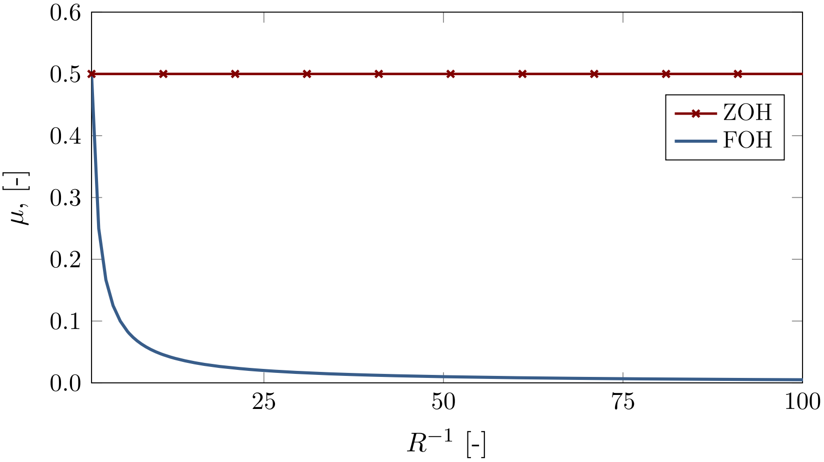

Fig. 12 Selection of the correction factor μ in a two-subsystem co-simulation, as a function of the step-size ratio and the extrapolation order when the subsystem with the smallest step-size has direct feedthrough

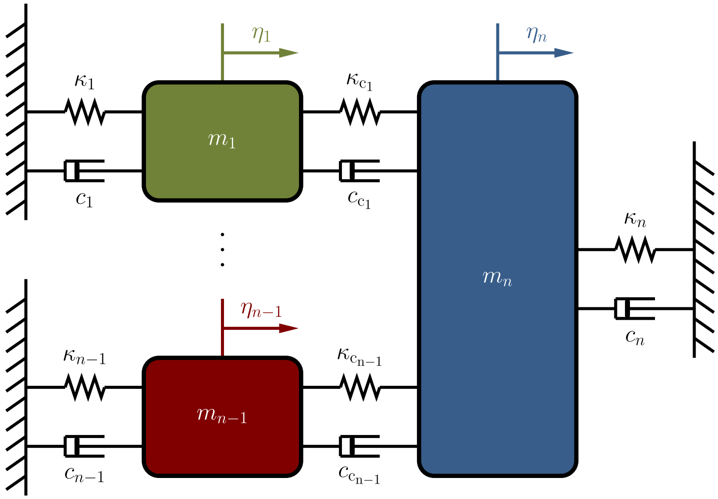

Fig. 13 A generic n-degree-of-freedom linear oscillator

4 Benchmark problems

The general validity of the correction coefficients μ and the applicability of the energy monitoring and correction methods put forward in Sect. 3 were verified with a set of co-simulation benchmark problems, which included nonlinear and multiphysics examples. In all cases, it was required that the product of their coupling variables delivered power units. The benchmarks were tested using single- and multi-rate co-simulation schemes.

Table 2 Combinations of system parameters used for the multiple linear oscillator

Case	n [-]	$m_{1\ldots n-1}$ [kg]	m_n [kg]	$\kappa_{1\ldots n-1}$ [N/m]	$\kappa_{\mathrm{c}1\ldots \mathrm{n}-1}$ [N/m]	κ_n [N/m]	$c_{1\ldots n-1}$ [Ns/m]	$c_{\mathrm{c}1\ldots \mathrm{n}-1}$ [Ns/m]	c_n [Ns/m]
1	3	1	1	10	100	1000	0	0	0
2	3	1	1	10	100	1000	0.1	0.1	0.1
3	5	1	1	10	100	1000	0	0	0
4	5	1	1	10	100	1000	0.1	0.1	0.1

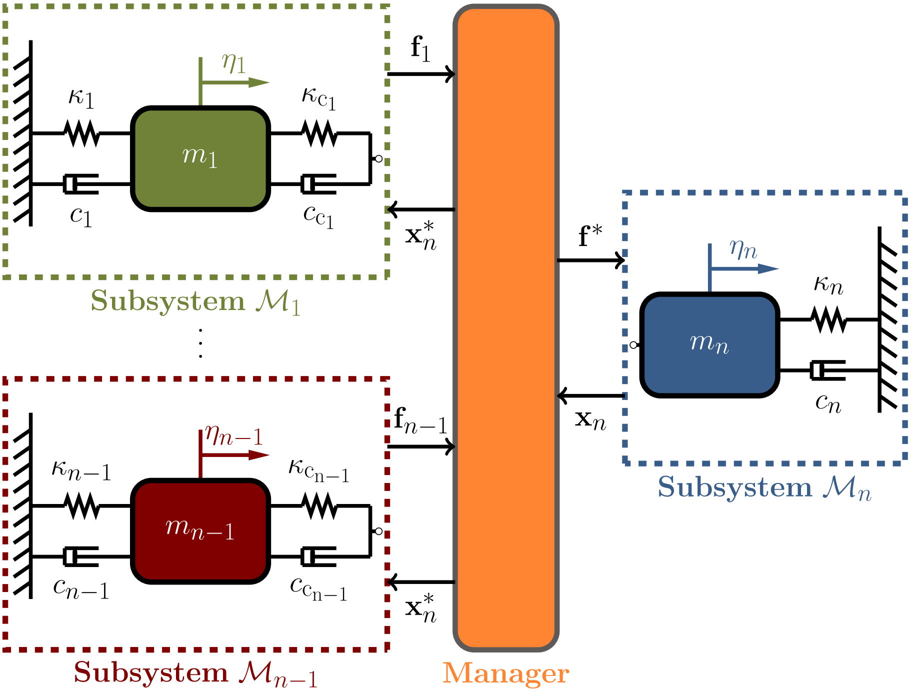

Fig. 14 The n-degree-of-freedom oscillator arranged following a force-displacement coupling scheme

4.1 Multiple linear oscillator

The first system used as a benchmark is a modification of the double linear oscillator described in Sect. 3.3, in which the first mass m_1 has been divided into $n-1$ masses, thus turning the system into the n-degree-of-freedom linear oscillator shown in Fig. 13.

The initial system displacements were set to 0 m for all masses. The velocities of the first $n-1$ masses were set to 100 m/s, whereas the velocity of the n-th mass was set to -100 m/s. Table 2 summarizes the system properties for all tested scenarios.

Figure 14 shows the oscillator arranged according to a force-displacement coupling scheme in which the subsystems $\mathcal{M}_1$ to $\mathcal{M}_{n-1}$ contain the information about their coupling spring-damper systems and evaluate the force at the interface. The algebraic sum of these coupling forces, here denoted as $\mathbf{f}$, is passed as input to the subsystem $\mathcal{M}_n$, which handles the integration of the dynamics equations of mass m_n. This subsystem returns as output its

Table 3 Combinations of system parameters used for the double nonlinear oscillator benchmark

Case	m_1 [kg]	m_2 [kg]	κ_1 [N/m]	κ_c [N/m]	κ_2 [N/m]	c_1 [Ns/m]	c_c [Ns/m]	c_2 [Ns/m]
1	1	1	10	100	1000	0	0	0
2	1	1	10	100	1000	0.001	0.001	0.001

position η_n and velocity $\dot{\eta}_n$, denoted here as $\mathbf{x}_n$. Direct feedthrough is found here in the first $n-1$ subsystems, which need their input $\mathbf{x}_n$ to evaluate their coupling forces.

4.2 Double nonlinear oscillator

The second benchmark example is another variation of the linear oscillator described in Sect. 3.3, in which the three springs and dampers can be replaced by nonlinear counterparts, thus transforming it into a double nonlinear oscillator. The nonlinear forces that are considered in this benchmark include

- *Nonlinear spring forces* that follow a square root function of spring elongation as

$$f = -\kappa \ \mathrm{sgn}(\eta) \ \sqrt{|\eta|} \tag{22}$$

  where sgn() is the sign function, η is the elongation of the spring, and κ is a constant stiffness coefficient.
- *Nonlinear damping forces* that follow a quadratic function of the damper elongation rate $\dot{\eta}$ as

$$f = -c \ \mathrm{sgn}(\dot{\eta}) \ |\dot{\eta}|^2 \tag{23}$$

  where c is a constant damping coefficient.

This nonlinear oscillator was tested in two different scenarios. In the first case, it is considered that all spring forces are nonlinear, as described in Eq. (22), and that damping is not present. The second case uses the nonlinear damping from Eq. (23) and linear spring forces. Table 3 summarizes the parameters used in the two scenarios proposed for this benchmark. The initial mass displacements and velocities were the same as those used for the linear oscillator: $\eta_1(0) = \eta_2(0) = 0$ m, $\dot{\eta}_1(0) = 100$ m/s and $\dot{\eta}_2(0) = -100$ m/s.

4.3 Coupled pendula

The third benchmark problem is a two-degree-of-freedom mechanical system comprised of two rigid pendula connected to each other at their free tips by a spring and a damper, as shown in Fig. 15.

The pendula consist of point masses m_1 and m_2, located at points 1 and 2, connected to two massless rods of lengths l_1 and l_2 and pinned to the ground at points A and B, respectively. Revolute joints exist at points A, B, 1, and 2, so that the rods and the spring-damper system can rotate freely around them. Gravity acts along the negative y axis, $g = 9.81$ m/s^2. The system parameters used in the simulations are detailed in Table 4. The initial angles between the rods and the negative y axis were set to $\theta_1(0) = \pi/4$ rad and $\theta_2(0) = \pi/18$ rad; the initial velocities were set to zero. In this initial configuration, the spring-damper is unloaded, thus its natural length l_0 is approximately 8.756 m.

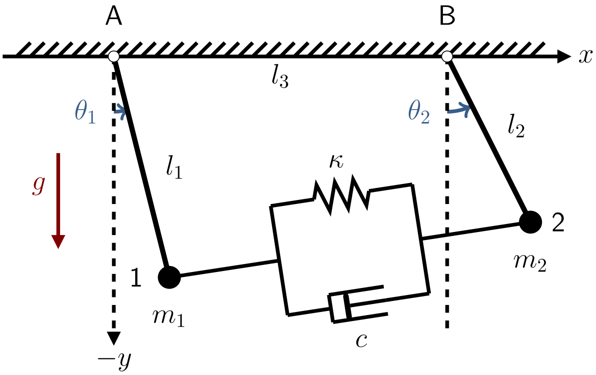

Fig. 15 Two simple pendula coupled by a spring-damper system

Table 4 Combinations of parameters used for the coupled pendula benchmark

Case	m_1 [kg]	m_2 [kg]	l_1 [m]	l_2 [m]	κ [N/m]	c [Ns/m]	A [m]	B [m]
1	1	2	2.5	3.5	6	0	(0, 0)	(7, 0)
2	1	2	2.5	3.5	50	0	(0, 0)	(7, 0)

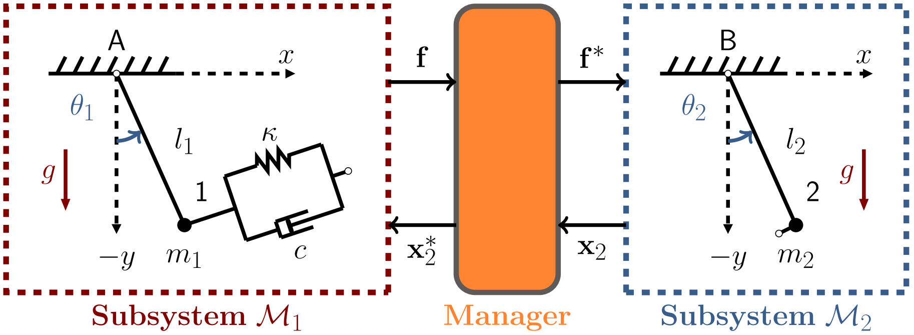

Fig. 16 A two-degree-of-freedom two pendula connected by a spring-damper at their tips arranged following a force-displacement coupling scheme

The system was divided following a force-displacement arrangement, shown in Fig. 16, for the co-simulation tests. The subsystem $\mathcal{M}_1$ integrates the dynamics of the first pendulum and evaluates the coupling force $\mathbf{f}$, which is passed on as input to the second subsystem. The subsystem $\mathcal{M}_2$ in turn solves the motion of the second pendulum and delivers the global positions (x_2, y_2) and velocities $(\dot{x}_2, \dot{y}_2)$ of its free tip 2, grouped in the array $\mathbf{x}_2$. The subsystem $\mathcal{M}_1$ is subjected to direct feedthrough. The corresponding coupling variables are

$$\mathbf{u}_{\mathcal{M}_1} = \left[\mathbf{x}_2^*\right] = \left[x_2^* \; y_2^* \; \dot{x}_2^* \; \dot{y}_2^*\right]^{\mathrm{T}} \qquad \mathbf{u}_{\mathcal{M}_2} = \left[\mathbf{f}^*\right] = \left[f_x^* \; f_y^*\right]$$
$$\mathbf{y}_{\mathcal{M}_1} = \left[\mathbf{f}\right] = \left[f_x \; f_y\right] \qquad \mathbf{y}_{\mathcal{M}_2} = \left[\eta_2 \; \dot{\eta}_2\right]^{\mathrm{T}} = \left[x_2 \; y_2 \; \dot{x}_2 \; \dot{y}_2\right]^{\mathrm{T}} \tag{24}$$

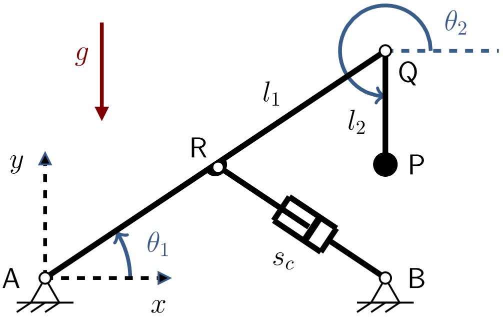

Fig. 17 A two-degree-of-freedom planar manipulator actuated by a hydraulic cylinder

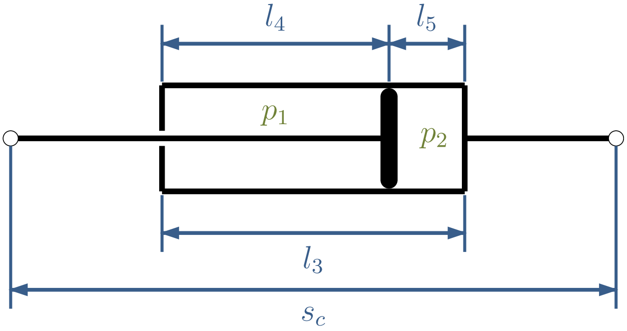

Fig. 18 Schematic of the hydraulic actuator

where again the superscript ()* denotes the modifications applied to the coupling variables by the co-simulation manager. The term **f** contains the x and y components of the coupling force exerted by the spring-damper.

In order to solve the equations of motion of the pendulum within each subsystem, different multibody system dynamics formulations could be used. Here, an augmented Lagrangian algorithm [6] was used in combination with the symplectic Euler integration formula. The dynamics of each subsystem is formulated as a system of differential algebraic equations (DAEs), which introduces a difference with respect to the oscillators presented so far, in which the dynamics was described by ordinary differential equations (ODEs).

4.4 Hydraulic crane

The last benchmark is a hydraulic crane that consists of a two-degree-of-freedom multiphysics system composed of a two-link robotic arm actuated by a hydraulic cylinder, as shown in Fig. 17. A similar system was initially described in [21], and modified versions were later included as benchmarks in [25] and [28]. The example here is the same as used in [25], although a different manoeuvre is performed in the numerical tests.

This example is a multiphysics assembly composed of two parts: a planar multibody system and a hydraulic circuit. The mechanical part consists of two rigid rods, connected to each other and to the ground by revolute joints, of lengths l_1 and l_2, respectively. Link 1 is pinned to the ground at point A and has a uniformly distributed mass m_1, whereas link 2 is massless. Two additional point masses m_2 and m_3 are located at Q and P, respectively.

The system is actuated with a hydraulic piston, shown in Fig. 18, and moves under gravity effects ($g = 9.81\ \mathrm{m/s}^2$ along the negative y axis). The internal pressures in the cylinder chambers are p_1 and p_2; the force exerted by the actuator is

$$f = a_p\left(p_2 - p_1\right) - c\dot{s}_c \tag{25}$$

where a_p is the cross section of the piston, c is the dissipation coefficient of the actuator, and s_c is the actuator length, which varies during motion. The dynamics of the hydraulic system can be described by the following system of differential equations

$$\dot{p}_1 = \frac{\beta_1}{a_p l_4}\left[a_p\dot{s}_c + a_i c_d\sqrt{\frac{2\left(p_{\mathrm{P}} - p_1\right)}{\rho}}\delta_{\mathrm{P}1} - a_o c_d\sqrt{\frac{2\left(p_1 - p_{\mathrm{T}}\right)}{\rho}}\delta_{\mathrm{T}1}\right] \tag{26}$$

$$\dot{p}_2 = \frac{\beta_2}{a_p l_5}\left[-a_p\dot{s}_c + a_o c_d\sqrt{\frac{2\left(p_{\mathrm{P}} - p_2\right)}{\rho}}\delta_{\mathrm{P}2} - a_i c_d\sqrt{\frac{2\left(p_2 - p_{\mathrm{T}}\right)}{\rho}}\delta_{\mathrm{T}2}\right] \tag{27}$$

where l_3 is the cylinder length, l_4 and l_5 are the lengths of the chambers on each side of the piston, a_i and a_o are the areas of the valves that connect the cylinder chambers to the pump and the tank in the hydraulic system (not represented for simplicity), ρ represents the density of the fluid, and c_d stands for the discharge coefficient of the valves. The quantities p_{P} and p_{T} are the hydraulic pressures at the pump and tank, respectively. The coefficients $\delta_{\mathrm{P}1}$, $\delta_{\mathrm{P}2}$, $\delta_{\mathrm{T}1}$, and $\delta_{\mathrm{T}2}$ are zero when the quantity inside the square root that precedes them is negative, otherwise they are equal to one. The terms β_1 and β_2 stand for the bulk modulus in each cylinder chamber and are evaluated as a function of fluid pressure as

$$\beta_i = \frac{1 + ap_i + bp_i^2}{a + 2bp_i}, \quad i = 1, 2 \tag{28}$$

where a and b are scalar constants that are fluid properties. Assuming that the two cylinder chambers have equal volume at the starting time of the simulation, the chamber lengths l_4 and l_5 are given by

$$l_4 = 0.5l_3 + s_c^0 - s_c \tag{29}$$

$$l_5 = 0.5l_3 + s_c - s_c^0 \tag{30}$$

where s_c^0 is the initial length of the actuator. The valve areas a_i and a_o are obtained as a function of the externally controlled displacement γ of a spool, which is an input of the system

$$a_i = 5 \cdot 10^{-4}\gamma \tag{31}$$

$$a_o = 5 \cdot 10^{-4}\left(1 - \gamma\right) \tag{32}$$

where $\gamma \in [0, 1]$. A more detailed description of the components of the hydraulic system, as well as the formulation used to model its behaviour can be found in [25]. The mechanical parameters used for this benchmark problem are summarized in Table 5.

At time $t = 0$, the initial angles of links 1 and 2 are $\theta_1(0) = \pi/6\,\mathrm{rad}$ and $\theta_2(0) = 3\pi/2\,\mathrm{rad}$. The system is initially in a state of static equilibrium. The force exerted by the actuator during the manoeuvre under study is regulated by the controlled valve, whose spool

Table 5 System parameters for the hydraulic crane benchmark

Length of link 1	l_1	1.0 m
Length of link 2	l_2	0.5 m
Cylinder length	l_3	0.442 m
Mass of link 1	m_1	200 kg
Point mass at Q	m_2	250 kg
Point mass at P	m_3	100 kg
Coordinates of fixed point B	$(x_\mathrm{B}, y_\mathrm{B})$	$\left(\sqrt{3}/2, 0\right)$ m
Friction coefficient	c	10^5 Ns/m
Piston area	a_p	$65 \cdot 10^{-4}$ m^2
Compressibility coefficient	a	$6.53 \cdot 10^{-10}$ Pa
Compressibility coefficient	b	$-1.19 \cdot 10^{-18}$
Discharge coefficient	c_d	0.67
Fluid density	ρ	850 kgm^{-3}
Hydraulic pressure at the pump	p_P	7.6 MPa
Hydraulic pressure at the tank	p_T	100 kPa

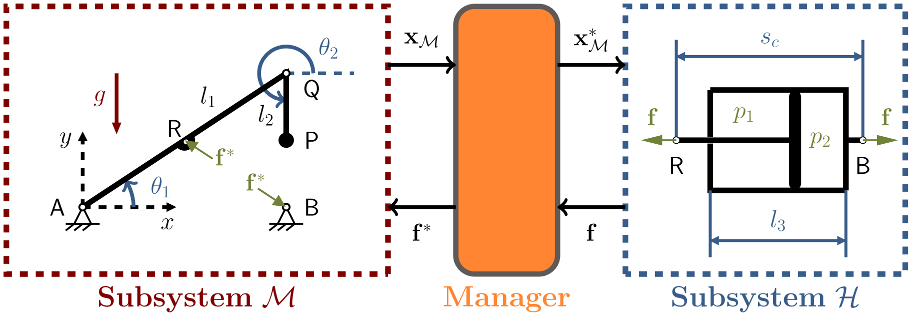

Fig. 19 A two-degree-of-freedom planar manipulator with a single hydraulic actuator arranged following a force-displacement coupling scheme

displacement γ follows a sinusoidal law with variable amplitude A and constant frequency $\omega = 2$ rad/s,

$$\gamma = \gamma_0 \left(1 - A \sin\left(2\pi\omega t\right)\right), \quad \text{where } A = \begin{cases} 0.1t\,, & 0\,\mathrm{s} \le t < 1\,\mathrm{s} \\ 0.1\,, & 1\,\mathrm{s} \le t < 8\,\mathrm{s} \\ 0.1\,(9-t)\,, & 8\,\mathrm{s} \le t < 9\,\mathrm{s} \\ 0\,, & t \ge 9\,\mathrm{s} \end{cases} \tag{33}$$

The spool displacement can vary between a completely closed ($\gamma = 0$) and a completely open ($\gamma = 1$) valve. Initially, the spool displacement was set to $\gamma_0 \approx 0.45435$, which corresponds to the static equilibrium of the system.

The different time scales required for the accurate solution of the multibody and hydraulic dynamics make it convenient to use multi-rate schemes in the co-simulation of this system. The integration step-size required for the mechanical subsystem is of the order of $h_\mathcal{M} = 1$ ms, while the hydraulic commonly requires steps around $h_\mathcal{H} = 0.1$ ms. The force-displacement coupling arrangement of the mechanical ($\mathcal{M}$) and hydraulic ($\mathcal{H}$) subsystems is shown in Fig. 19. The subsystem $\mathcal{M}$ integrates the multibody system dynamics using as

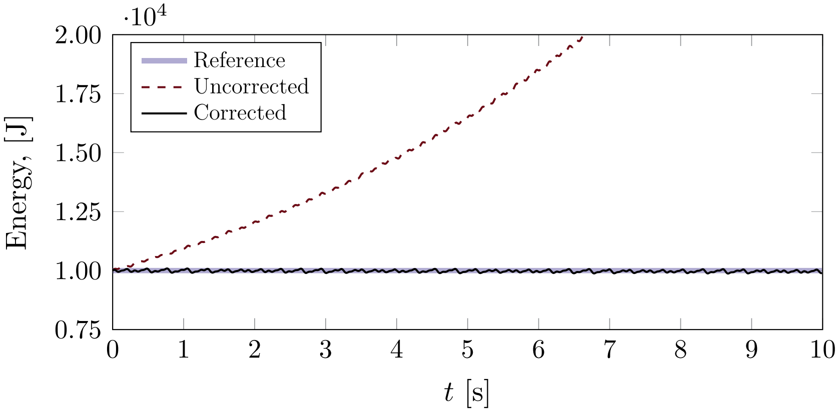

Fig. 20 Double linear oscillator: Mechanical energy in the multi-rate co-simulation of case 1 with ZOH extrapolation: $H = h_{\mathcal{M}1} = 1$ ms, $h_{\mathcal{M}2} = 0.1$ ms, $\mu = 0.725$, the correction force was limited to 100 % of the original coupling force

input the force $\mathbf{f}$ exerted by the piston and returns as output $\mathbf{x}_{\mathcal{M}}$ the displacement s_c and rate $\dot{s}_c$ of the cylinder. The hydraulic subsystem $\mathcal{H}$, in turn, integrates its dynamics to obtain the pressures in the cylinder and evaluates the hydraulic force using Eq. (25). In this configuration, the hydraulic subsystem $\mathcal{H}$ features direct feedthrough. Thus, the coupling variables for this example are

$$\begin{aligned} \mathbf{u}_{\mathcal{H}} &= \left[\mathbf{x}^*_{\mathcal{M}}\right] = \left[s_c{}^* \ \dot{s}^*_c\right]^{\mathrm{T}} & \qquad \mathbf{u}_{\mathcal{M}} &= \left[\mathbf{f}^*\right] = \left[f^*\right] \\ \mathbf{y}_{\mathcal{H}} &= \left[\mathbf{f}\right] = \left[f\right] & \qquad \mathbf{y}_{\mathcal{M}} &= \left[\mathbf{x}_{\mathcal{M}}\right] = \left[s_c \ \dot{s}_c\right]^{\mathrm{T}} \end{aligned} \tag{34}$$

5 Results

The explicit co-simulation of the linear oscillator described in Sect. 3, as well as the rest of benchmark problems introduced in Sect. 4, were used to assess the ability of the residual power δP to indicate co-simulation quality. The numerical experiments were also used to verify the energy correction method discussed in Sect. 3.2. Both single-rate and multi-rate Jacobi schemes were evaluated, using the different sets of physical parameters referred in Tables 1–5. In all cases, the integration step-sizes of the subsystems matched the size of their communication step with the manager.

For each example, the residual power δP was monitored first during a co-simulation run without corrective actions. In a second round of simulations, the energy correction method was evaluated; in these, the coefficient μ was adjusted using the relations obtained for the linear oscillator shown in Eqs. (20) and (21). In most examples it was necessary to introduce an additional correction by means of the removal of the accumulated energy difference, as done in Eq. (17), in order to deliver optimal results.

The results obtained with the double oscillator in Sect. 3.3 have confirmed that an almost exact removal of the excess energy is possible for this linear system. Figure 20 illustrates the effect of the energy correction in the multi-rate co-simulation of case 1 in Table 1. A correction coefficient $\mu = 0.725$ was used, determined with Eq. (20) for a step-size ratio $R = 10$ and ZOH extrapolation. The effect of the energy correction is reflected in the predicted

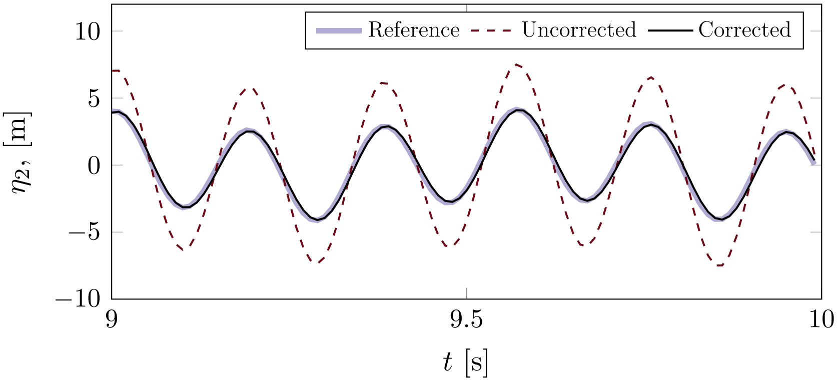

Fig. 21 Double linear oscillator: Displacement η_2 of mass m_2 in the multi-rate co-simulation of case 1 with ZOH extrapolation: $H = h_{\mathcal{M}_1} = 1$ ms, $h_{\mathcal{M}_2} = 0.1$ ms, $\mu = 0.725$, the correction force was limited to 100 % of the original coupling force

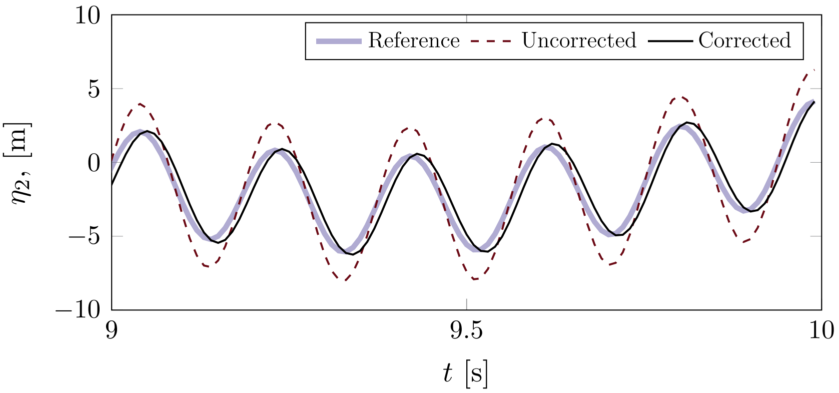

Fig. 22 Double linear oscillator: Displacement η_2 of mass m_2 in the single-rate co-simulation of case 7: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms, $\mu = 0.5$, the correction force was limited to 100 % of the original coupling force

system motion, shown in Fig. 21 for the same scenario. The increase in motion amplitude that resulted from the accumulation of spurious energy has been removed and the corrected solution matches with the analytical reference. It must be noted that a small delay with respect to the reference remained in the results; this effect has already been observed with other energy-based correction algorithms [13].

The values of μ calculated with Eqs. (20) and (21) were used as reference for the co-simulation of the rest of cases in Table 1 and the benchmark examples. Figure 22 shows the effect of the energy correction on the position η_2 of the mass m_2 of the double linear oscillator in case 7 with a corrective factor $\mu = 0.5$, corresponding to single-rate co-simulation. Again, the motion amplitude matches that of the reference and a small delay can be observed in the results. Figure 23 compares the mechanical energy of the system with and without corrections. In this case, the mass m_2 in the subsystem that receives the corrected force is an order of magnitude smaller than the one in the other subsystem m_1. This affects negatively the ability of the method to bring the energy level down to that of the reference. More effective removal of the excess energy can be achieved by introducing the integral

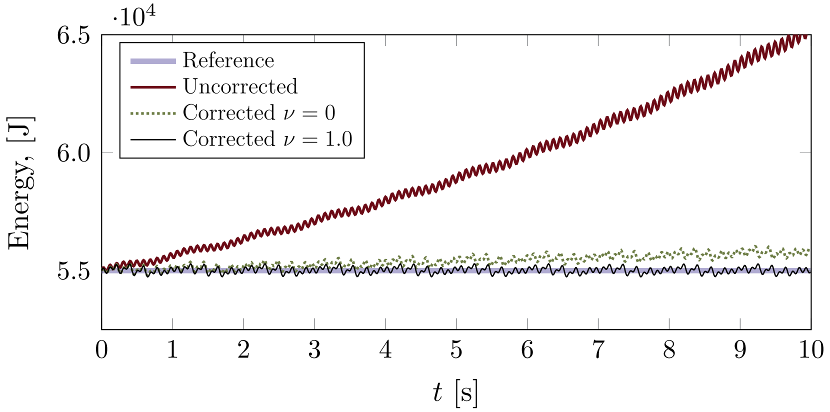

Fig. 23 Double linear oscillator: Energy in the single-rate co-simulation of case 7: $H = h_{\mathcal{M}1} = h_{\mathcal{M}2} = 1$ ms, $\mu = 0.5$, the correction force was limited to 100 % of the original coupling force

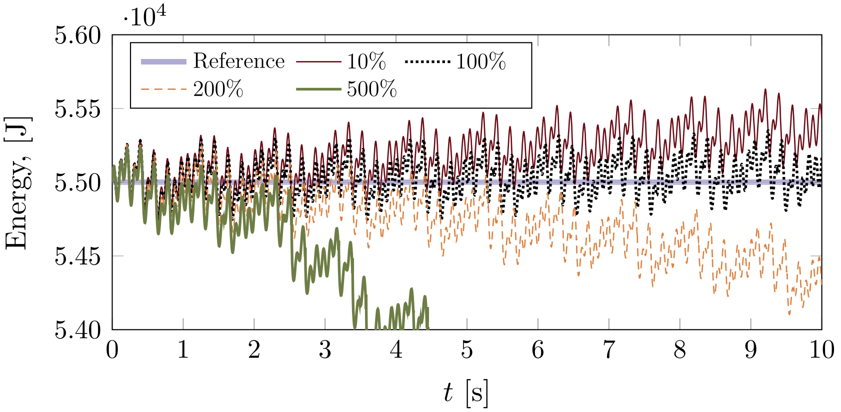

Fig. 24 Double linear oscillator: Effect of the limit to the correction force on mechanical energy in single-rate co-simulation of case 7: $H = h_{\mathcal{M}1} = h_{\mathcal{M}2} = 1$ ms, $\mu = 0.5$, $\nu = 0.25$. The force limit is expressed as a percentage of the original coupling force

term in Eq. (17). Fig. 24 illustrates the need to set a limit on the maximum correction force exerted at the interface. Large values of correction can be detrimental to simulation accuracy, whereas insufficient modifications can fail to remove completely energy errors. In all problems used in this paper, the maximum admissible correction force has not exceeded 100% of the uncorrected coupling force. The combination of an appropriate integral factor ν and a force limit is problem-dependent, and the development of an approach to select both systematically remains an open task within this research.

The proposed correction approach can also be used when the co-simulation environment includes more than two systems. This is the case with the multiple linear oscillator in Sect. 4.1, in which the left-hand side consists of a set of $n - 1$ subsystems that pass their coupling force to the manager. This example was simulated using $n = 3$ and $n = 5$ subsystems. Figures 25 and 26 confirm that it is possible to extend the use of the correction method

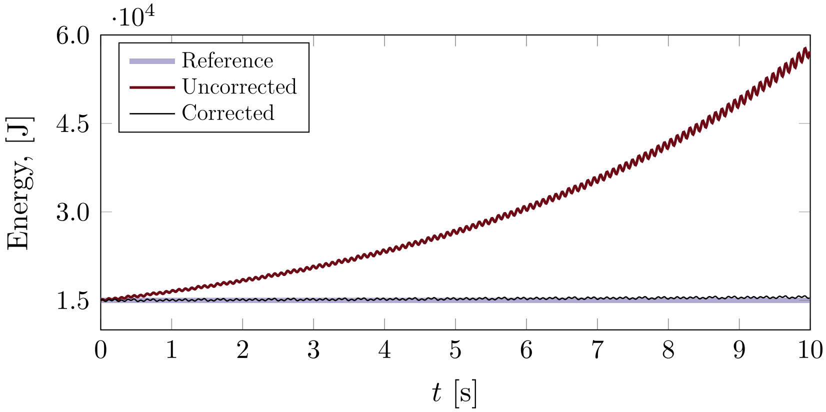

Fig. 25 Multiple linear oscillator: Mechanical energy in single-rate co-simulation of case 1: $n = 3$, $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = h_{\mathcal{M}_3} = 1$ ms, $\mu_1 = \mu_2 = 0.5$, the correction force was limited to 25 % of the original coupling force

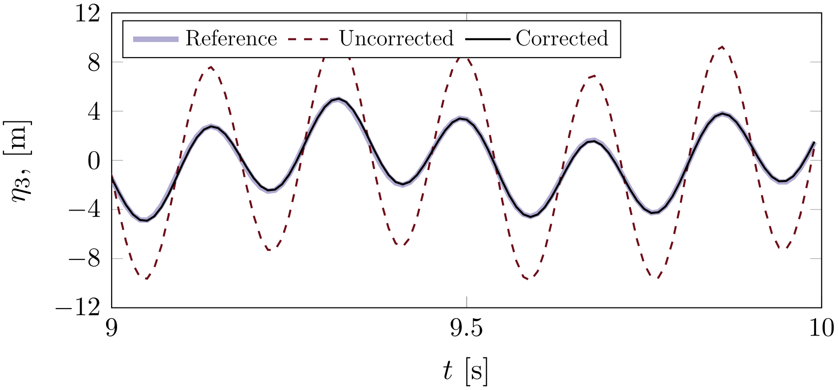

Fig. 26 Multiple linear oscillator: Displacement η_3 in single-rate co-simulation of case 1: $n = 3$, $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = h_{\mathcal{M}_3} = 1$ ms, $\mu_1 = \mu_2 = 0.5$, the correction force was limited to 25 % of the original coupling force

to systems composed of more than two subsystems, taking into account the structure of the power bonds present between the subsystems. In this example, a single correction force was applied to subsystem n, as shown in Fig. 14.

The co-simulation of the nonlinear systems in Sect. 4 required the use of an integral correction in most cases. Figure 27 shows the mechanical energy of the double nonlinear oscillator in Sect. 4.2 that results from the use of different correction factors μ in the absence of the integral term ν. The energy of the uncorrected system ($\mu = 0$) grows consistently with time. The factor that corresponds to single-rate co-simulation, $\mu = 0.5$, initially performs an acceptable control of the system energy, but eventually is not able to appropriately handle the nonlinear behaviour of the benchmark. Raising μ above 0.5 results in over-corrections, rendering the system stable at the cost of dissipating too much energy.

The introduction of an integral term ν resulted in the ability to remove energy deviations in spite of the nonlinear behaviour of the test problem, as confirmed by Fig. 28. Different

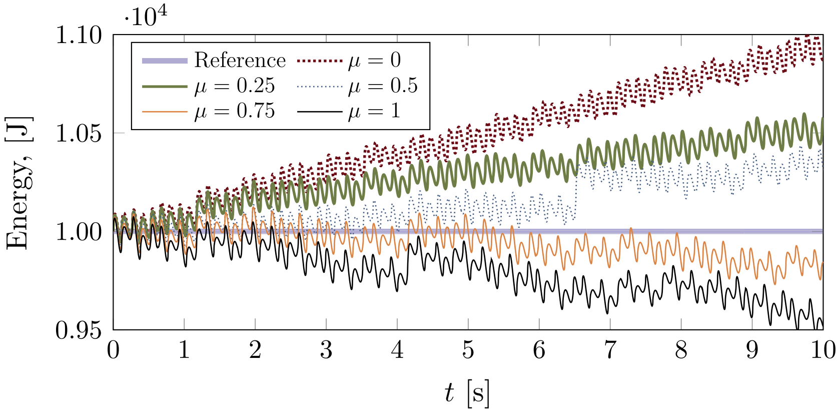

Fig. 27 Double nonlinear oscillator: Effect of the correction coefficient on system energy in a single-rate co-simulation of the case 1: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms, $\nu = 0$, the correction force was limited to 100 % of the original coupling force

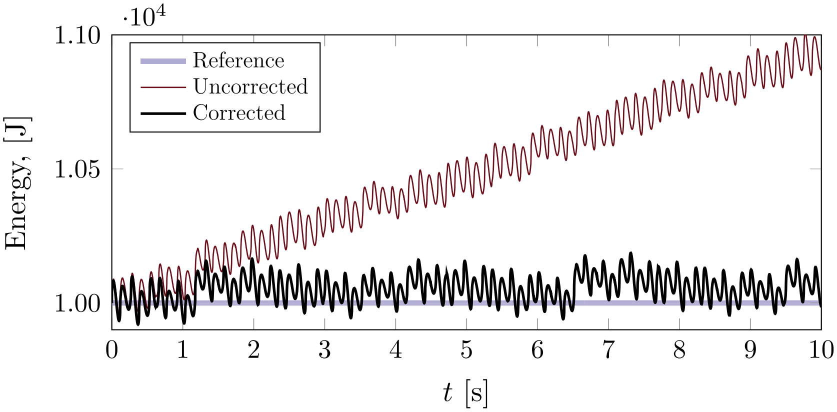

Fig. 28 Double nonlinear oscillator: Energy in single-rate co-simulation of case 1: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms, $\mu = 0.5$, $\nu = 0.25$, the correction force was limited to 100 % of the original coupling force

values of the integral term were tested, with $\nu = 0.25$ being the optimal value to keep the co-simulation stable and accurate. The displacement η_2 with this configuration is represented in Fig. 29. In case 2, in which nonlinear damping was present, the severity of the energy deviations with respect to the reference solution was reduced and it was possible to correct the motion without including integral terms, as shown in Fig. 30.

Numerical experiments with the nonlinear oscillator also confirmed the role of the residual power, δP, as an indicator of co-simulation quality with nonlinear systems. Figure 31 shows the time-history of the residual power during the uncorrected co-simulation of case 1. Although the values vary over time, the average of the signal is above zero, which indicates the consistent growth of the energy stored in the system that can be observed in Figs. 27 and 28.

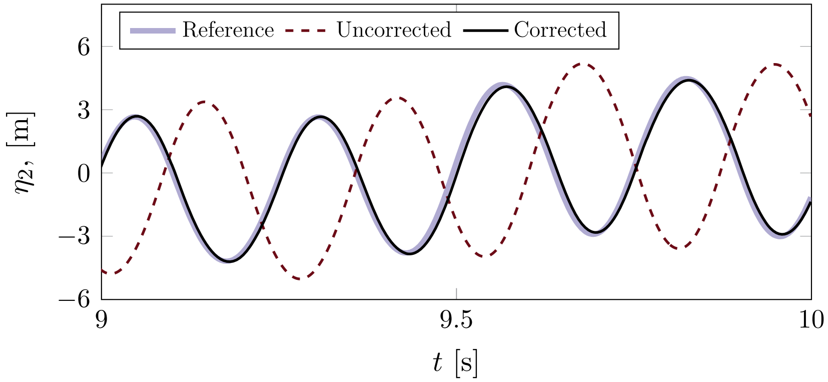

Fig. 29 Double nonlinear oscillator: Displacement η_2 in the single-rate co-simulation of case 1: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms, $\mu = 0.5$, $\nu = 0.25$, the correction force was limited to 100 % of the original coupling force

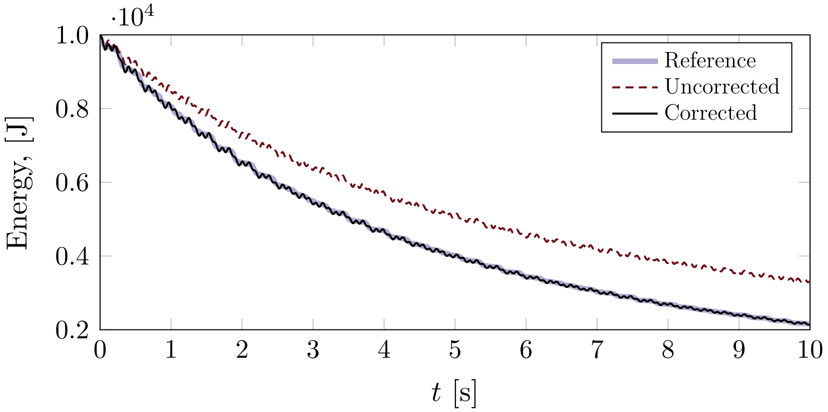

Fig. 30 Double nonlinear oscillator: Energy in the single-rate co-simulation of case 2: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms, $\mu = 0.5$, the correction force was limited to 100 % of the original coupling force

Similar results regarding the ability of the proposed method to correct energy deviations in the system were obtained with the rest of co-simulation benchmarks. Figure 32 compares the x_2 coordinate during the simulation of the coupled pendulum example with and without energy correction. Even though the deviation from the reference solution introduced by the co-simulation scheme was relatively small, the correction method was able to bring the results closer to the reference motion. Figure 33 illustrates the relevance of correction: the spurious energy introduced at the interface is small, but it accumulates steadily over time, degrading the quality of the results as the simulation proceeds. The residual power, shown in Fig. 34, indicates this energy behaviour: Its value, albeit small, remains above zero during motion, which corresponds to a slow but steady increase of the system energy. The proposed correction method removed this excess energy; it must be noted that an integral term was necessary to achieve adequate results.

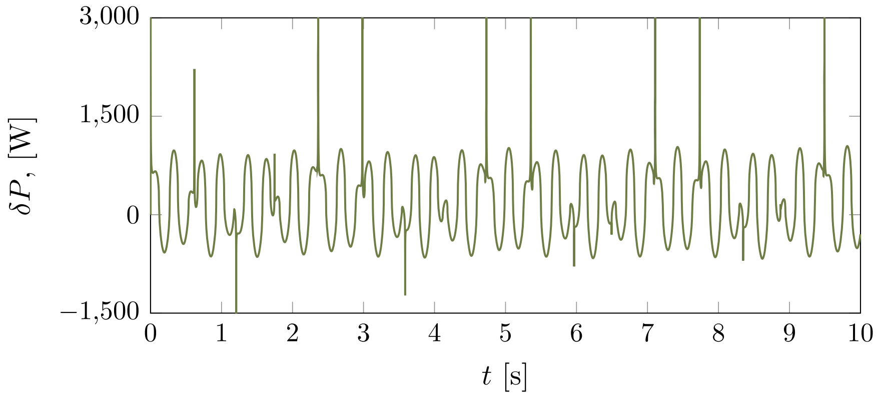

Fig. 31 Double nonlinear oscillator: Residual power δP in the uncorrected single-rate co-simulation of case 1: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms

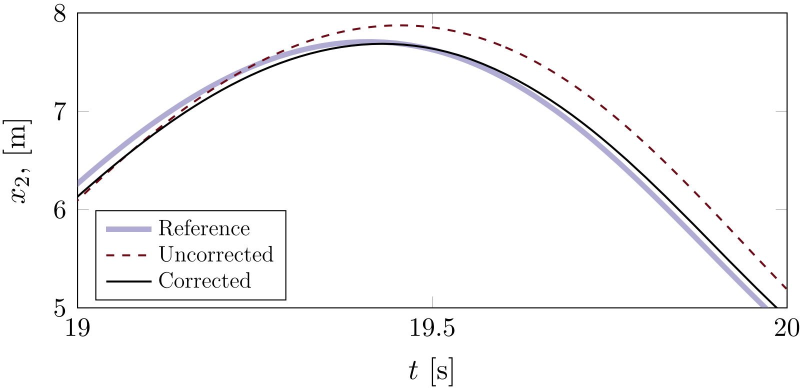

Fig. 32 Coupled pendulum: Coordinate x_2 in the single-rate co-simulation of case 1. $H = h_{\mathcal{M}1} = h_{\mathcal{M}2} = 1$ ms, $\mu = 0.5$, $\nu = 1.0$, the correction force was limited to 100 % of the original coupling force

The last benchmark example, the hydraulic crane, is a multi-physics system in which the mechanical and hydraulic subsystems present very different time scales. Such a problem benefits in terms of efficiency from the use of multi-rate co-simulation schemes. In this example, due to the significant dissipation present at the hydraulic cylinder, the deviations of the uncorrected co-simulated dynamics from the reference solution remained relatively small in terms of both motion and energy, as long as the macro-steps for each subsystems, $h_{\mathcal{M}}$ and $h_{\mathcal{H}}$, were kept small enough to ensure the stability of the integration. Figure 35 shows the results obtained with FOH extrapolation and $h_{\mathcal{M}} = 10$ ms and $h_{\mathcal{H}} = 0.25$ ms, which correspond to a step-size ratio $R = 1/40$. In this configuration, the actuator displacement follows closely, within a 2- mm margin, the reference motion. Unlike in the previous examples, the coupling errors at the discrete-time interface have removed energy from the system, instead of increasing the energy level. The energy correction, nonetheless, improves again the accuracy of the results.

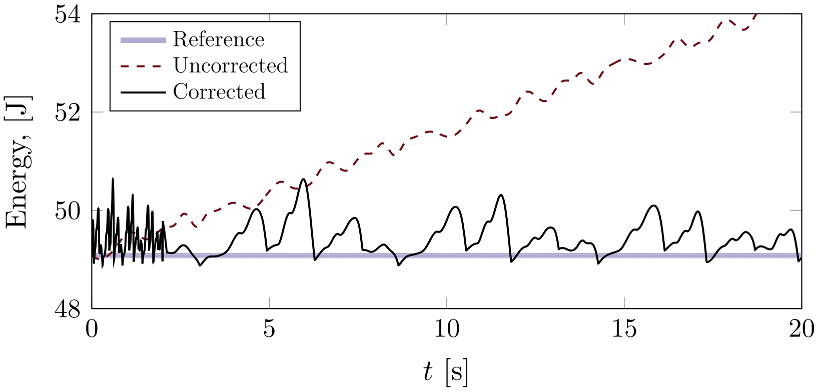

Fig. 33 Coupled pendulum: Energy in the single-rate co-simulation of case 1. $H = h_{\mathcal{M}1} = h_{\mathcal{M}2} = 1$ ms, $\mu = 0.5$, $\nu = 1.0$, the correction force was limited to 100 % of the original coupling force

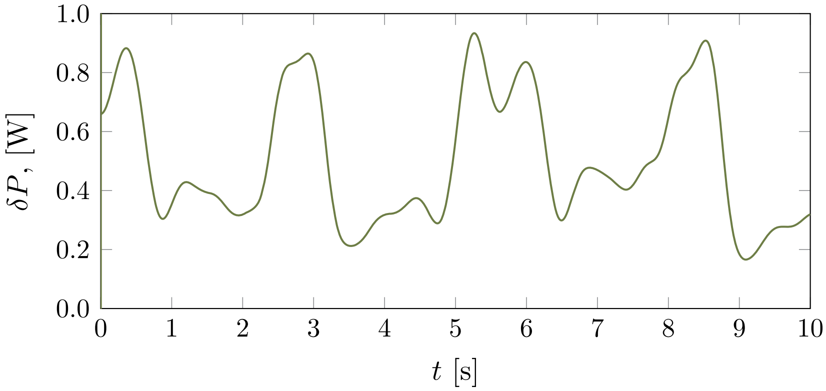

Fig. 34 Coupled pendulum: Residual power δP in the uncorrected single-rate co-simulation of case 1: $H = h_{\mathcal{M}_1} = h_{\mathcal{M}_2} = 1$ ms

This example confirmed as well the usefulness of the residual power δP as an indicator of the co-simulation quality. Figure 36 shows the actuator length s_c during the uncorrected co-simulation of the manoeuvre described by Eq. (33), using $h_{\mathcal{M}} = 11.5$ ms, $h_{\mathcal{H}} = 0.1$ ms and FOH extrapolation. The co-simulated solution shows an oscillation between $t = 3$ s and $t = 5$ s, approximately, which does not correspond to the actual system motion, as evidenced by the comparison with the reference solution. In a practical application, however, a reference solution will not be available in most cases and such a comparison would not be feasible, so that it would be unclear whether the oscillation is caused by a numerical artifact or if it actually corresponds to the true system behaviour. In fact, the co-simulation recovers from the oscillation and remains stable, partly because of the dissipative behaviour of the cylinder. However, the time-history of the residual power contains a clear spike at the time during which the oscillation occurs, as can be seen in Fig. 37, which permits the identification of the oscillations as a result of numerical errors and thus labelling the simulation as unreliable.

It is important to mention that in all simulated examples, it was necessary to select an upper limit for the magnitude of the correction force that was applied to the subsystems via

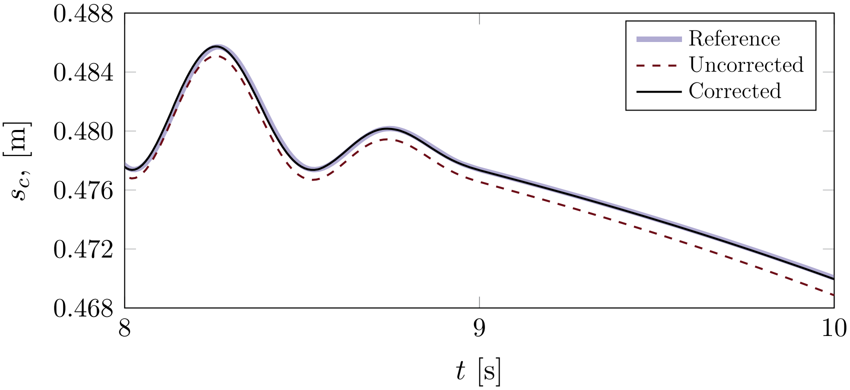

Fig. 35 Hydraulic crane: Cylinder displacement s_c in multi-rate co-simulation. $H = h_{\mathcal{M}} = 10$ ms, $h_{\mathcal{H}} = 0.25$ ms, $\mu = 0.0125$, $\nu = 0$, FOH extrapolation, the correction force was limited to 10 % of the original coupling force or 100 N

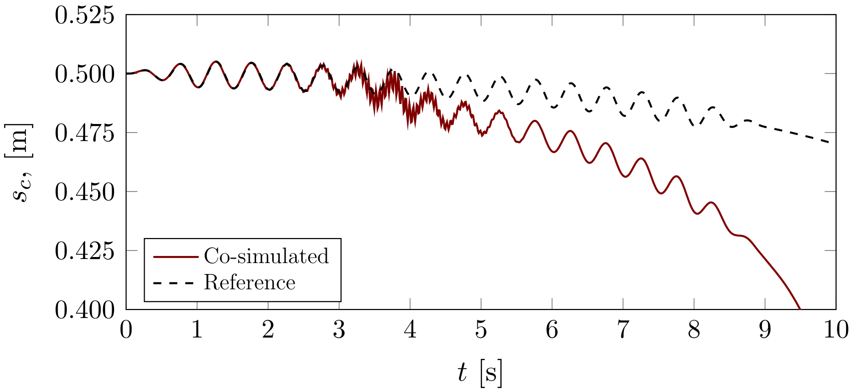

Fig. 36 Hydraulic crane: Cylinder displacement s_c in the multi-rate co-simulation of the system motion: $H = h_{\mathcal{M}} = 11.5$ ms and $h_{\mathcal{H}} = 0.1$ ms with FOH extrapolation for the inputs of the hydraulic subsystem, the correction force was limited to 10 % of the original coupling force or 100 N

Eq. (14) or Eq. (17). Using these equations, the theoretical correction force $\mathbf{f}^{\text{corr}}$ can grow indefinitely at points when the velocity used as the coupling variable, $\mathbf{v}$, approaches zero. In all tested cases, the magnitude of the correction force $\mathbf{f}^{\text{corr}}$ was kept below that of its uncorrected counterpart $\mathbf{f}$ at each instant in time, although the stability and accuracy of the results could be improved in some scenarios by lowering this limit. The development of an automated way to tune beforehand these force caps, as well as the integral term gain ν, for each particular simulation scenario, remains a currently open line of research.

6 Conclusions

Explicit schemes are frequently used in co-simulation applications when efficient performance is a requirement. Among these, fixed-step Jacobi schedules enable the parallelization

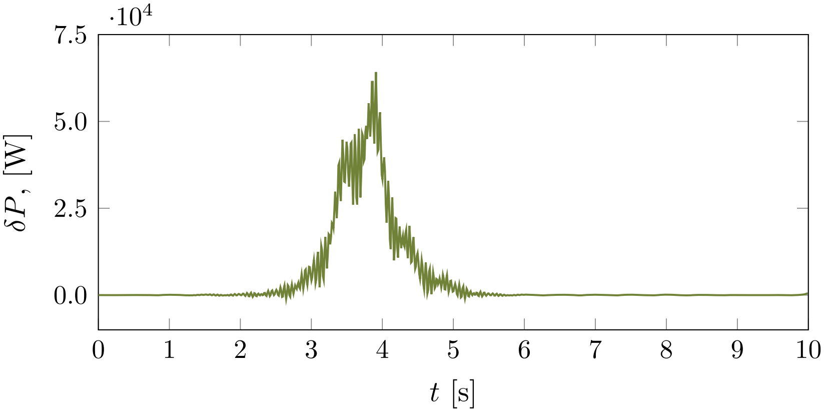

Fig. 37 Hydraulic crane: Residual power δP in the multi-rate co-simulation of the system motion: $H = h_{\mathcal{M}} = 11.5$ ms and $h_{\mathcal{H}} = 0.1$ ms with FOH extrapolation for the inputs of the hydraulic subsystem

of solver execution and are particularly suitable for their integration in demanding environments, such as those that demand real-time behaviour. These explicit schemes, however, are known to suffer from stability and accuracy issues, especially when the subsystem dynamics are subjected to direct feedthrough. In these cases, it is necessary to assess co-simulation quality during runtime to ensure the reliability of the obtained results.

Residual power, an indicator that can be evaluated from the coupling variables in force-displacement co-simulation configurations, can be used to monitor the quality of numerical integration and enable actions to prevent the simulation from delivering incorrect results and becoming unstable in extreme cases. This magnitude was also used to develop an energy-correction method to remove the energy inconsistencies caused by the discrete-time co-simulation interface. The method introduces a corrective action in the coupling variables and requires limited knowledge of the subsystem internals. Its behaviour was demonstrated in the numerical integration of a linear oscillator, a commonly used benchmark problem in the co-simulation literature, and further verified with a set of additional test examples. These included nonlinear and multiphysics systems; multi-rate co-simulation schemes were tested beside single-rate ones. The results showed that residual power can be used to gain insight into the co-simulation quality, and that the proposed energy correction method can be employed to remove the spurious variations in system energy caused by discontinuities and delays at discrete co-simulation interfaces.

Acknowledgements F. González acknowledges the support of the Ministry of Economy of Spain through the Ramón y Cajal research program, contract no. RYC-2016-20222. This work was partially financed by the Government of Galicia under grants ED431B2016/031 and ED431F2021/04 and the Ministry of Economy of Spain through project TRA2017-86488-R "Técnicas de co-simulación en tiempo real para bancos de ensayo en automoción".

Funding Note Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Authors also acknowledge the University of A Coruña/CISUG for the funding of the open access charges.

Declarations

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

References

1. FMI – Functional Mock-up Interface (2021). https://fmi-standard.org/
2. Andersson, C.: Methods and tools for co-simulation of dynamic systems with the Functional Mock-up Interface. Ph.D. thesis, Lund University (2016)
3. Arnold, M.: How to verify worst case asymptotic error bounds for co-simulation algorithms. In: González, F., Cuadrado, J. (eds.) Proceedings of COSIM2021, an International Symposium on Co-simulation and Solver Coupling in Dynamics, Ferrol, Spain, pp. 48–49 (2021)
4. Arnold, M., Clauss, C., Schierz, T.: Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation v2.0. Arch. Mech. Eng. **60**(1), 75–94 (2013). https://doi.org/10.2478/meceng-2013-0005
5. Barbosa, T.P., da Silva, L.A.R., Pujatti, F.J.P., Gutiérrez, J.C.H.: Hydraulic hybrid passenger vehicle: Fuel savings possibilities. Mechanics Based Design of Structures and Machines, 1–19 (2020). https://doi.org/10.1080/15397734.2020.1714447, Early access
6. 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), 183–195 (1988). https://doi.org/10.1016/0045-7825(88)90085-0
7. Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: Macro-step-size selection and monitoring of the coupling errof for weak coupled subsystems in the frequency-domain. In: International Conference on Computational Methods for Coupled Problems in Science and Engineering - Ibiza, Spain (2013)
8. Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: A nearly energy-preserving coupling element for holistic weak-coupled system co-simulations. In: NAFEMS World Congress 2013, Salzburg, Austria (2013)
9. Burger, M., Steidel, S.: Local extrapolation and linear-implicit stabilization in a parallel coupling scheme. In: IUTAM Symposium on Solver-Coupling and Co-Simulation, pp. 43–56. Springer, Berlin, Germany (2019). https://doi.org/10.1007/978-3-030-14883-6_3
10. Busch, M.: Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. Z. Angew. Math. Mech. **96**(9), 1061–1081 (2016). https://doi.org/10.1002/zamm.201500196
11. Chen, W., Ran, S., Wu, C., Jacobson, B.: Explicit parallel co-simulation approach: analysis and improved coupling method based on H-infinity synthesis. Multibody Syst. Dyn. **52**(3), 255–279 (2021). https://doi.org/10.1007/s11044-021-09785-x
12. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: A survey. ACM Comput. Surv. **51**(3), 1–33 (2018). https://doi.org/10.1145/3179993
13. 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**, 172–188 (2019). https://doi.org/10.1016/j.mechmachtheory.2018.09.007
14. González, F., Naya, M.Á., Luaces, A., González, M.: On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. Multibody Syst. Dyn. **25**(4), 461–483 (2011). https://doi.org/10.1007/s11044-010-9234-7
15. Haid, T., Stettinger, G., Watzenig, D., Benedikt, M.: A model-based corrector approach for explicit co-simulation using subspace identification. In: Proceedings of the 5th Joint International Conference on Multibody System Dynamics, Lisbon, Portugal (2018)
16. Kraft, J., Klimmek, S., Meyer, T., Schweizer, B.: Implicit co-simulation and solver-coupling: efficient calculation of interface-Jacobian and coupling sensitivities/gradients. J. Comput. Nonlinear Dyn. (2021). https://doi.org/10.1115/1.4051823 Accepted manuscript
17. Kübler, R., Schiehlen, W.: Modular simulation in multibody system dynamics. Multibody Syst. Dyn. **4**(2/3), 107–127 (2000). https://doi.org/10.1023/a:1009810318420
18. Lacoursière, C., Härdin, T.: FMI Go! A simulation runtime environment with a client server architecture over multiple protocols. In: Proceedings of the 12th International Modelica Conference, Prague, Czech Republic (2017). https://doi.org/10.3384/ecp17132653

19. Li, P., Yuan, Q.: Influence of coupling approximation on the numerical stability of explicit co-simulation. J. Mech. Sci. Technol. **34**(6), 2289–2298 (2020). https://doi.org/10.1007/s12206-020-0504-x
20. Meyer, T., Kraft, J., Schweizer, B.: Co-simulation: Error estimation and macro-step size control. J. Comput. Nonlinear Dyn. **16**(4), 041002 (2021). https://doi.org/10.1115/1.4048944
21. Naya, M., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: Comparison with simplified and co-integration approaches. Arch. Mech. Eng. **58**(2), 223–243 (2011). https://doi.org/10.2478/v10180-011-0016-4
22. Newmark, N.M.: A method of computation for structural dynamics. J. Eng. Mech. Div., ASCE **85**(EM3), 67–94 (1959). https://doi.org/10.1061/jmcea3.0000098
23. Oberschelp, O., Vöcking, H.: Multirate simulation of mechatronic systems. In: Proceedings of the IEEE International Conference on Mechatronics, 2004. IEEE, Istanbul, Turkey (2004). https://doi.org/10.1109/icmech.2004.1364473
24. Olivier, B., Verlinden, O., Kouroussis, G.: Effect of applied force cosimulation schemes on recoupled vehicle/track problems. Multibody Syst. Dyn. **50**(4), 337–353 (2020). https://doi.org/10.1007/s11044-020-09748-8
25. 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**, 52–72 (2018). https://doi.org/10.1016/j.mechmachtheory.2018.04.016
26. Peiret, A., González, F., Kövecses, J., Teichmann, M.: Co-simulation of multibody systems with contact using reduced interface models. J. Comput. Nonlinear Dyn. **15**(4), 041001 (2020). https://doi.org/10.1115/1.4046052
27. Rahikainen, J., González, F., Naya, M.Á.: An automated methodology to select functional co-simulation configurations. Multibody Syst. Dyn. **48**(1), 79–103 (2020). https://doi.org/10.1007/s11044-019-09696-y
28. Rahikainen, J., González, F., Naya, M.Á., Sopanen, J., Mikkola, A.: On the cosimulation of multibody systems and hydraulic dynamics. Multibody Syst. Dyn. **50**(2), 143–167 (2020). https://doi.org/10.1007/s11044-020-09727-z
29. Rodríguez, B., Sanjurjo, E., Tranchero, M., Romano, C., González, F.: Thermal parameter and state estimation for digital twins of e-powertrain components. IEEE Access **9**, 97384–97400 (2021). https://doi.org/10.1109/access.2021.3094312
30. 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), 607–620 (2017). https://doi.org/10.1007/s00366-016-0492-8
31. Sadjina, S., Pedersen, E.: Energy conservation and coupling error reduction in non-iterative co-simulations. Eng. Comput. **36**(4), 1579–1587 (2020). https://doi.org/10.1007/s00366-019-00783-4
32. 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), 345–373 (2007). https://doi.org/10.1007/s11044-007-9076-0
33. 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
34. Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. Arch. Appl. Mech. **84**(12), 1739–1769 (2014). https://doi.org/10.1007/s00419-014-0883-5
35. Schweizer, B., Lu, D.: Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. Multibody Syst. Dyn. **34**(2), 129–161 (2015). https://doi.org/10.1007/s11044-014-9422-y
36. Zhang, R., Zhang, H., Zanoni, A., Wang, Q., Masarati, P.: A tight coupling scheme for smooth/non-smooth multibody co-simulation of a particle damper. Mech. Mach. Theory **161**, 104181 (2021). https://doi.org/10.1016/j.mechmachtheory.2020.104181

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