Eigenstructure assignment and compensation of explicit co-simulation problems

Iacopo Tamellin, Dario Richiedei, Borja Rodríguez, Francisco González

This is a post-peer-review, pre-copyedit version of an article published in Mechanism and Machine Theory. The final authenticated version is available online at: https://doi.org/10.1016/j.mechmachtheory.2022.105004.

Abstract

Co-simulation is an effective and versatile way to determine the forward-dynamics behaviour of complex engineering applications. In co-simulation setups, the overall system dynamics is split into several subsystems that evolve in time separately. This makes it possible to use modelling and integration methods that can be tailored to the specific nature and behaviour of each of them. Co-simulation subsystems coordinate their execution by means of information exchanges through a discrete-time interface. In some cases, this limited exchange of data can cause accuracy and stability issues in the simulation process, especially when explicit coupling schemes are employed. Correction algorithms are then required to ensure the accuracy of the obtained results. This paper provides insight into the structure of explicit co-simulation problems, revealing the effect of input extrapolation at the discrete-time interface between subsystems. The resulting system equations are formulated in terms of control theory expressions, which can be then used to develop compensation solutions to correct the perturbations introduced at the co-simulation interface. The compensator architecture is chosen to ensure the eigenstructure assignability condition, which has been ad-hoc developed in this paper. These aspects are illustrated here in the explicit co-simulation of linear mechanical systems.

Keywords: Co-simulation, Explicit coupling schemes, Pole placement, Eigenstructure assignment, Extrapolation, Linear mechanical systems

1 Introduction

1.1 Motivations of the paper

The predictive simulation of complex engineering applications is a challenging task that often requires the coordinated use of modelling and solution approaches for different physical domains. Such system-level simulation needs to consider the dynamics of the components that are involved in the application, as well as their interactions with each other and the environment. Developments in hardware and software have made it possible to deal with systems of considerable complexity using a single simulation tool, following a *monolithic approach*, e.g., [40, 28]. Monolithic solutions employ a common solver for every component in the setup; this solver typically has access to all the necessary modelling and implementation details used to define the assembly. *Co-simulation*, on the other hand, is based on treating each component separately as a *subsystem* with its own dynamics description and solver [19]. Subsystems exchange information at certain instants in time, known as *communication points*, by means of a set of *coupling variables*. In the interval between two consecutive communication points, or *macro-step*, the integration of the dynamics of each subsystem proceeds without further knowledge of the evolution of its environment. This makes co-simulation a modular approach to perform the numerical integration of complex system dynamics; at the same time, it requires the intervention of an *orchestrator* or *co-simulation manager* to coordinate the execution of the different solvers.

Co-simulation subsystems can be coupled following a wide variety of schemes; in general, these can be categorized into two main groups: implicit (iterative) schemes, and explicit or non-iterative [41]. Implicit co-simulation schemes repeat the integration of the subsystems between communication points until a certain convergence criterion is satisfied. Explicit schemes do not allow this iteration. In general, implicit co-simulation features better stability properties [25] and it is often assumed to yield more accurate results as well. It must be mentioned, however, that stability and accuracy are not equivalent concepts. A co-simulated system can be made stable, for instance, introducing dissipation in its numerical integration process, thus compromising the accuracy of the results. Keeping co-simulation not only stable, but accurate as well is a critical objective when designing a co-simulation environment.

The use of implicit co-simulation is not always possible in practical applications. Sometimes the repetition of macro-steps may be computationally too expensive; this can be the case, for instance, if real-time execution of the simulation code is a requirement. Repeating the integration

of subsystem dynamics between communication points also requires subsystem *rollback*, i.e., resetting the subsystem state to a previously calculated point in time. Not every subsystem can perform rollback; physical components in hybrid co-simulation setups are an example of this. These reasons make explicit schemes a frequently adopted option in practical implementations of co-simulation.

Explicit co-simulation solutions, however, may suffer from instability and inaccuracy issues that stem from the discrete-time interface that connects the subsystems [20]. The identification of the sources of this degraded behaviour and the formulation of means to address it are necessary to ensure the correct operation of practical co-simulation setups that follow explicit schemes.

1.2 State of the art

The discrete-time interface between co-simulation subsystems gives rise to discontinuities and delays in the exchange process of the coupling variables [13]. These, in turn, lead to the introduction of errors in the numerical integration process, particularly when direct feedthrough is present in one or more subsystems [6]. Because subsystem solvers must proceed with their integration between communication points without knowledge of the state of the rest of the system, it may occur that some subsystem inputs need to be approximated, e.g., via polynomial extrapolation; these approximated inputs often do not match the true values actually delivered by the other subsystems [38]. Implicit co-simulation schemes can reduce these errors carrying out iterative input evaluations; in explicit ones, conversely, this is not possible and the dynamics simulation, if left uncorrected, accumulates a deviation from its expected theoretical behaviour. In order to keep explicit co-simulation setups stable and accurate, it is necessary to remove or mitigate this deviation.

Input extrapolation [13, 29] is a frequently used means to alleviate the effect of discontinuities in the coupling variables at the co-simulation interface. However, the selection of an appropriate extrapolation technique is often problem-dependent, and finding the strategy that delivers the best possible results can become a challenging task [21, 33]. The extrapolation method can also be adjusted as the simulation progresses, based on previously experienced system behaviour [8]. When additional information about the subsystems is available, other so-

lutions to approximate the values of the coupling variables can be proposed, for instance if the subsystems provide the partial derivatives of their output with respect to their state and input [24], or if the dynamics of some subsystems are known in detail, so that model-based coupling techniques can be used [43, 32, 23].

Adjusting the macro step-size to match the frequency of the coupled dynamics is another possibility to avoid unstable behaviour of explicit co-simulation environments [10]. This adjustment can also be done using energy-based indicators to determine the step-size [39]. In real-time applications, however, it may be difficult to modify the macro step-size during runtime, especially if the co-simulation includes physical components.

Another family of techniques to improve co-simulation quality includes those methods that rely on performing modifications on the coupling variables to remove artefacts introduced at the interface. These modifications often incorporate elements from control theory to determine the corrective actions that need to be carried out on the subsystem inputs, such as passivity controllers, e.g., [11, 20, 38, 14]. The explicit analysis of the pole-zero structure of the problems, put forward in this paper, is less frequent in the literature about co-simulation of mechanical systems. An advantage of input-correction solutions is that they do not require to modify the macro step-size or the subsystem internals; their applicability, however, is limited by the nature of the coupling variables and, in physical subsystems, by the configuration and properties of the sensors and actuators that are mounted on them.

1.3 Contributions of the paper

This paper puts forward a model of explicit Jacobi co-simulated systems. The proposed formulation is developed in terms of the perturbation that co-simulation introduces with respect to its theoretical monolithic counterpart. Such framework enables one to determine the effect of the signal exchange at the co-simulation interface between subsystems, revealing the way in which the system eigenstructure, i.e., the eigenvalues (the poles) and the eigenvectors (the mode shape), is perturbed. This perturbation, in turn, degrades co-simulation accuracy and may lead in extreme cases to instability.

A novel compensation algorithm, based on the eigenstructure assignment control technique for linear systems, is proposed in this paper. A compensator that exploits the feedback of dis-

placements, speeds, and accelerations is designed to make the eigenstructure of the compensated co-simulated system match that of the theoretical system. Additionally, the poles (the "latent roots") due to the time-delay introduced at the coupling interface are accounted for in the compensator synthesis to make them stable. A relevant feature of the proposed scheme is that it does not require time-consuming, trial-and-error tuning since the computation of the compensator gains is based on a rigorous physical base and is, therefore, automatically performed from the eigenstructure of the monolithic system, which can be either directly known or approximated by means of subsystem identification techniques, e.g., [22].

The proposed method is applicable to linear mechanical systems and its effectiveness is assessed through the numerical simulation of the motion of a two-mass system, a frequently used benchmark in the co-simulation literature. Besides, the method is extended to the case of co-simulation with subsystems that feature more than one degree of freedom. Hence, the effectiveness of the method for systems with more degrees of freedom is corroborated by the numerical tests performed on a four-mass system model.

2 Analysis of the coupling of subsystems

2.1 Representation of the coupling and general overview of a co-simulated system

Let us consider a typical co-simulation setup, composed of two weakly coupled linear time-invariant subsystems, as shown in Fig. 1. The coupling variables are exchanged among the subsystems through a co-simulation manager at each communication point. The macro step-size between two consecutive communication points is T_s. The manager coordinates the numerical integration of the subsystems and adjusts their inputs extrapolating them from the values provided at previous communication points.

Thanks to the linearity of the subsystems, a representation in the Laplace s-domain can be adopted (s is the complex Laplace variable). The Laplace transform is useful to model the co-simulation manager as well, by means of the extrapolator transfer functions $\mathrm{D}(s)$ from the

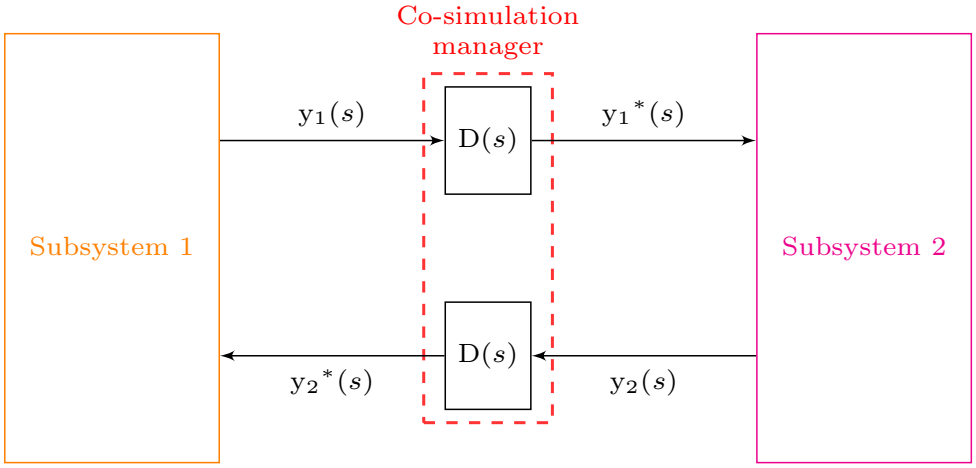


Figure 1: General block scheme of a co-simulated system.

signals sent (y_1, y_2) to the ones received (y_1^*, y_2^*):

$$\begin{aligned} y_1^*(s) &= D(s)\, y_1(s) \\ y_2^*(s) &= D(s)\, y_2(s) \end{aligned} \tag{1}$$

Without loss of generality, in this paper it is assumed that the same $D(s)$ is applied to both connections, although the proposed theory admits the use of a different extrapolation for each subsystem.

A general technique for data extrapolation is to use a polynomial to fit from past samples. If a zero-order polynomial is employed, i.e., the extrapolation is done by keeping the signal constant during the time-step, then the extrapolator is denoted as the zero-order hold (ZOH), whose transfer function can be expressed as [9] :

$$D(s) = \frac{1 - e^{-sT_s}}{sT_s} \tag{2}$$

The transfer function in Eq. (2) is frequently used in the co-simulation literature, e.g., [11, 10]. Indeed, for small enough macro step-sizes, it has been shown to be an acceptable assumption [11].

Higher-order polynomials are sometimes adopted, such as the first-order hold (FOH) or the second-order hold (SOH) [16, 9], which exploit linear or quadratic extrapolations, respectively. Recently, non-polynomial, higher-order extrapolators based on the H_∞ synthesis have been proposed as well [14]. Besides the extrapolation model, $D(s)$ should often include time-delay terms due to sending or receiving latencies (respectively t_i and t_r), whose transfer functions are expo-

nentials, e^{-st_i} or e^{-st_r} [43]. Once the co-simulation manager has been properly modelled in the s-domain, the methods proposed in this paper can be applied regardless of the specific transfer functions $\mathrm{D}(s)$ that describe its behaviour.

2.2 General representation of perturbations due to co-simulation

Figure 1 reveals that a co-simulation can be seen as a closed-loop interconnection of two open-loop subsystems. Compared to the monolithic formulation, the transfer function of the whole system should account for the presence of $\mathrm{D}(s)$ that perturbs it. First of all, it can be expected that such a perturbation might deteriorate the simulation accuracy. Second, since $\mathrm{D}(s)$ usually features at least an exponential term e^{-sT_s}, the resulting transfer function is not rational; hence an infinite number of poles is introduced [30, 4].

To clearly evaluate the perturbation introduced by the co-simulation manager, let us first consider the analytical model of the N-DOF (degrees of freedom) linear time-invariant multi-body system, in its monolithic representation described by the set of N ordinary differential equations (ODE) of motion:

$$\mathbf{M}\ddot{\boldsymbol{x}}(t) + \mathbf{C}\dot{\boldsymbol{x}}(t) + \mathbf{K}\boldsymbol{x}(t) = \boldsymbol{f_e}(t) \tag{3}$$

where $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{N\times N}$ are the mass, damping, and stiffness matrices. The generalized, independent displacements are collected in vector $\boldsymbol{x} \in \mathbb{R}^N$ and $\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}$ are the velocities and accelerations respectively, while $\boldsymbol{f_e} \in \mathbb{R}^N$ are the generalized external forces.

The Laplace transformation of Eq. (3) yields the introduction of the dynamic stiffness matrix $\mathbf{G_t}(s) = \left(s^2\mathbf{M} + s\mathbf{C} + \mathbf{K}\right) \in \mathbb{C}^{N\times N}$ [44] of the monolithic system, that is in practice the theoretical model:

$$\mathbf{G_t}(s)\,\mathbf{x}(s) = \mathbf{f_e}(s) \tag{4}$$

where vectors $\mathbf{x}(s)$ and $\mathbf{f_e}(s)$ are the Laplace transform of $\boldsymbol{x}(t)$ and $\boldsymbol{f_e}(t)$ respectively. The dynamic behaviour of the monolithic system is completely described by its eigenstructure, i.e., the $2N$ eigenpairs $(\lambda_i, \mathbf{w}_i)$ with $\lambda_i \in \mathbb{C}$ denoting an eigenvalue (also denoted as a pole) and $\mathbf{w}_i \in \mathbb{C}^N$ its related eigenvector (also denoted as the mode shape), that are the solutions of the

eigenproblem:

$$\mathbf{G_t}\left(\lambda_i\right)\mathbf{w}_i = \mathbf{0}, \quad i = 1, ..., 2N \tag{5}$$

In the case of a complex-conjugate pair of eigenvalues (hereafter j denotes the complex number in this paper), the i-th pair is $\lambda_i = -\xi_i\omega_{d,i} \pm \omega_{d,i}j$ where the i-th damped natural frequency is $\omega_{d,i} = \omega_{n,i}\sqrt{1-\xi_i^2}$, ξ_i is its damping ratio and $\omega_{n,i}$ the natural frequency. Its related eigenvector $\mathbf{w}_i$ is the i-th mode shape and it describes the pattern of vibration through the spatial variation of the amplitude of the motion across the system, in terms of normalized displacements, when the system freely evolves at its damped natural frequency. Eigenvalues and eigenvectors are of fundamental importance to set both the free and the forced response of the system.

If the system is co-simulated as in Fig. 1, its equations of motion are perturbed by the presence of the extrapolator $\mathrm{D}\left(s\right)$, and Eq. (4) is no longer a correct representation. Hence, the dynamic stiffness matrix of the co-simulated system can be notionally written as the sum of the theoretical transfer function and a non-rational perturbation matrix, $\mathbf{\Delta G_c}\left(s\right)$, that depends on the features of the coupling, i.e., on the exchanged variables and on $\mathrm{D}\left(s\right)$:

$$\left(\mathbf{G_t}\left(s\right) + \mathbf{\Delta G_c}\left(s\right)\right)\mathbf{x}\left(s\right) = \mathbf{f_e}\left(s\right) \tag{6}$$

In this paper it is assumed that the time-discrete interface in co-simulation schemes introduces larger errors than the one caused by the subsystem numerical integration schemes, which is a reasonable assumption in explicit co-simulation of systems with direct feedthrough [39, 20]. The integration scheme used in the remainder of the paper, both inside the subsystems and for monolithic implementations, is the single-step, explicit symplectic Euler formula. This simplifies the interpretation of the coupling error. For instance, if implicit integrators were used, it would be necessary to consider the effect of input extrapolation on the convergence of their iteration process. The presence of $\mathbf{\Delta G_c}\left(s\right)$ deteriorates the accuracy of the simulations and, in the worst case, leads to diverging results, i.e., to instability.

The first issue is related to the eigenproblem:

$$\left(\mathbf{G_t}\left(\tilde{\lambda}_i\right) + \mathbf{\Delta G_c}\left(\tilde{\lambda}_i\right)\right)\tilde{\mathbf{w}}_i = \mathbf{0} \tag{7}$$

The solution of the eigenproblem of the co-simulated system yields $2N$ perturbed eigenpairs $\left(\tilde{\lambda}_i, \tilde{\mathbf{w}}_i\right)$ that do not match those of the monolithic system, where $\tilde{\lambda}_i$ is the i-th perturbed eigenvalue and $\tilde{\mathbf{w}}_i$ is the perturbed eigenvector. The perturbation of the $2N$ primary roots (i.e., the "physical ones"), the so called "pole spillover", might shift them to the right half plane (i.e., with positive real parts), thus destabilizing the co-simulation process. Besides changing the primary roots, the presence of the exponential terms in $\mathrm{D}(s)$ leads to a transcendental form of the characteristic equation of the co-simulated system, that features an infinite number of roots, the so-called "secondary" or "latent" roots.

In open loop systems, the presence of such latent roots is not critical since those lie at $s = -\infty$; unfortunately this is not the case of co-simulated systems. Indeed, since a co-simulated system behaves as a closed-loop interconnection of subsystems, these roots migrate from $-\infty$ and move towards the low frequency region and might also cross the imaginary axis. This discussion shows that the eigenstructure analysis reveals *a priori* and with a systematic approach the inaccuracy and instability that might arise in co-simulated systems.

Setting $\mathrm{D}(s) = 1$ leads to the monolithic model since $\mathbf{\Delta G_c}(s) = \mathbf{0}$. It should also be noted that, usually, the larger T_s the more the eigenstructure is perturbed, since $\mathrm{D}(s)$ differs from the ideal unitary gain model in a broader range of frequencies in terms of both amplitude and phase.

2.3 Explanatory example: the two-mass system

2.3.1 Two-mass system: theoretical and co-simulated model

A detailed explanation of the general theory developed is proposed in this section through a simply connected, two-mass linear vibrating system. Such a system is widely employed in the literature as a benchmark problem for co-simulation [21, 9, 20, 14, 27, 26, 13, 42].

The model of the monolithic system in the Laplace domain, assuming that no external forces are applied, is:

$$\left(s^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} + s \begin{bmatrix} c_1 + c_c & -c_c \\ -c_c & c_c + c_2 \end{bmatrix} + \begin{bmatrix} k_1 + k_c & -k_c \\ -k_c & k_c + k_2 \end{bmatrix} \right) \begin{Bmatrix} \mathrm{x}_1(s) \\ \mathrm{x}_2(s) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (8)$$

where m_1, m_2 are the two masses, and c_1, c_2 and k_1, k_2 denote the damping and stiffness parameters of the connections between masses 1 and 2 and the ground. The two masses are coupled through a spring k_c and a damper c_c. By exploiting the notation used in Eq. (4), the following transfer function is obtained:

$$\mathbf{G_t}(s) = s^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} + s \begin{bmatrix} c_1 + c_c & -c_c \\ -c_c & c_c + c_2 \end{bmatrix} + \begin{bmatrix} k_1 + k_c & -k_c \\ -k_c & k_c + k_2 \end{bmatrix} \tag{9}$$

It should be noted that $\mathbf{G_t}(s)$ can be obtained by assembling the transfer functions of the two decoupled subsystems, respectively denoted as $\mathbf{G_1}(s)$ and $\mathbf{G_2}(s)$, and the transfer function of the coupling, i.e., $\mathbf{G_c}(s)$. Hence, the knowledge of the monolithic model of the system is not mandatory. For example, in the case of the two-mass system under consideration it is possible to define:

$$\begin{aligned} \mathbf{G_1}(s) &= s^2 m_1 + s c_1 + k_1 \\ \mathbf{G_2}(s) &= s^2 m_2 + s c_2 + k_2 \\ \mathbf{G_c}(s) &= s c_c + k_c \end{aligned} \tag{10}$$

Hence, the monolithic system transfer function is indirectly obtained by assembling the matrices in Eq. (10) as follows:

$$\mathbf{G_t}(s) = \begin{bmatrix} \mathbf{G_1}(s) + \mathbf{G_c}(s) & -\mathbf{G_c}(s) \\ -\mathbf{G_c}(s) & \mathbf{G_2}(s) + \mathbf{G_c}(s) \end{bmatrix} \tag{11}$$

The co-simulated system is shown in Fig. 2. A force-displacement coupling approach is employed, hence the coupling variables are $f_c = -k_c(x_1 - x_2^*) - c_c(\dot{x_1} - \dot{x_2}^*)$ and x_2. This selection of coupling variables causes subsystem 1 to suffer from direct feedthrough, because its input x_2^* needs to be known to evaluate its output f_c. It can be shown that the force-displacement configuration results in worse co-simulation stability and accuracy properties than its displacement-displacement counterpart [6, 38], which makes it more challenging from the point of view of compensation.

The effect of the discrete-time co-simulation interface and the extrapolation methods, as

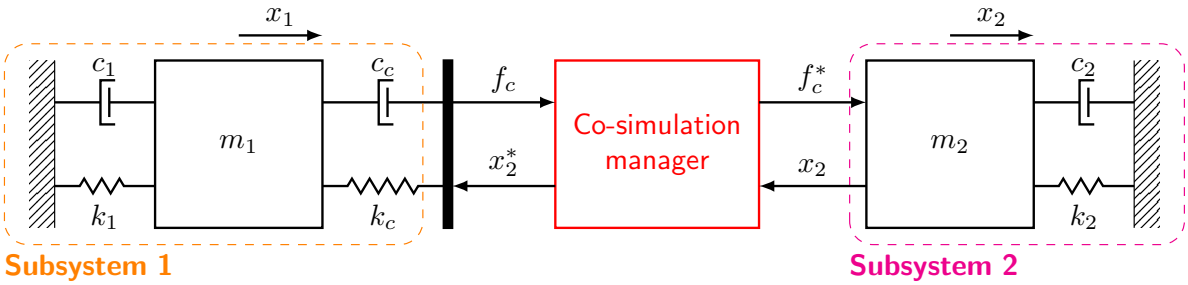


Figure 2: Co-simulated two-mass system with force-displacement coupling.

defined in Eq. (1), results in

$$\begin{aligned} \mathrm{f}_c^*(s) &= \mathrm{D}(s)\,\mathrm{f}_c(s) \\ \mathrm{x}_2^*(s) &= \mathrm{D}(s)\,\mathrm{x}_2(s) \end{aligned} \tag{12}$$

where $\mathrm{f}_c^*(s)$ and $\mathrm{x}_2^*(s)$ are respectively the extrapolated coupling force and displacement of mass 2. Hence, the Laplace domain equations describing the dynamic behaviour of the co-simulated two-mass system are:

$$\begin{cases} \left(s^2 m_1 + s(c_1+c_c) + k_1 + k_c\right)\mathrm{x}_1(s) = (sc_c + k_c)\mathrm{D}(s)\,\mathrm{x}_2(s) \\ \left(s^2 m_2 + sc_2 + k_2\right)\mathrm{x}_2(s) = (sc_c + k_c)\mathrm{D}(s)\,\mathrm{x}_1(s) - (sc_c + k_c)D^2(s)\,\mathrm{x}_2(s) \end{cases} \tag{13}$$

Finally, the model of the co-simulated two-mass system can be written as done in Eq. (6), as:

$$\left(\mathbf{G_t}(s) + \begin{bmatrix} 0 & (sc_c + k_c)(1 - \mathrm{D}(s)) \\ (sc_c + k_c)(1 - \mathrm{D}(s)) & (sc_c + k_c)\left(D^2(s) - 1\right) \end{bmatrix}\right) \begin{Bmatrix} \mathrm{x}_1(s) \\ \mathrm{x}_2(s) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{14}$$

Hence, the perturbation matrix introduced by the co-simulation interface is:

$$\mathbf{\Delta G_c}(s) = \begin{bmatrix} 0 & (sc_c + k_c)(1 - \mathrm{D}(s)) \\ (sc_c + k_c)(1 - \mathrm{D}(s)) & (sc_c + k_c)\left(\mathrm{D}^2(s) - 1\right) \end{bmatrix} \tag{15}$$

2.3.2 Two-mass system: numerical analysis

Let us consider the same parameters employed in [20] for the two-mass system, i.e., $m_1 = m_2 = 1$ kg, $k_1 = 10$ N/m, $k_2 = 1000$ N/m and $k_c = 100$ N/m. As for the damping, two scenarios are considered:

- scenario 1: $c_1 = c_2 = c_c = 0$ Ns/m;
- scenario 2: $c_1 = c_2 = c_c = 0.1$ Ns/m.

The macro-step size of the co-simulation is set equal to $T_s = 1$ ms and ZOH extrapolation is assumed.

Let us first consider the undamped system (scenario 1). The eigenpairs $(\lambda_i, \mathbf{w}_i)$ of the monolithic system are summarized in Table 1 together with the modal parameters and with the eigenpairs $\left(\tilde{\lambda}_i, \tilde{\mathbf{w}}_i\right)$ of the co-simulated system. The poles of the co-simulated system are here computed by using the 6-th order Padé approximation of the exponential term that appears in the ZOH transfer function, see Eq. (2).

The Padé approximant yields a rational transfer function that approximates the exponential term e^{-sT_s} by [45]:

$$e^{-sT_s} \approx \frac{\sum_{i=0}^{m_p} p_i \left(sT_s\right)^i}{\sum_{i=0}^{n_p} q_i \left(sT_s\right)^i} \tag{16}$$

where p_i and q_i are defined as follows:

$$p_i = (-1)^i \frac{(2n_p - i)! n_p!}{(2n_p)! i! \left(n_p - i\right)!} \quad q_i = \frac{(2n_p - i)! n_p!}{(2n_p)! i! \left(n_p - i\right)!}, \quad i = 1, ..., n_p \tag{17}$$

A common choice is to set the order of the Padé approximation $m_p = n_p$ (obviously, m_p cannot be greater than n_p to obtain a proper transfer function). High-order Padé approximations produce transfer functions with clustered poles; since such pole configurations are very sensitive to perturbations, such as round-off errors, it is usually suggested using the Padé approximation with $n_p \leq 10$ [18].

The poles listed in Table 1 evidence that co-simulation causes spillover on the primary poles of the monolithic system λ_i, whose counterparts in the co-simulated system are $\tilde{\lambda}_i$, as shown in Fig. 3. The latent roots due to co-simulation at higher natural frequencies are separately shown in Fig. 4.

The pole analysis reveals that the co-simulated pole pair $\tilde{\lambda}_{3,4}$ lies on the right-half of the complex plane, thus making the co-simulated system unstable (i.e., diverging even with bounded input), as will be confirmed by the numerical experiments carried out in Section 5.2.

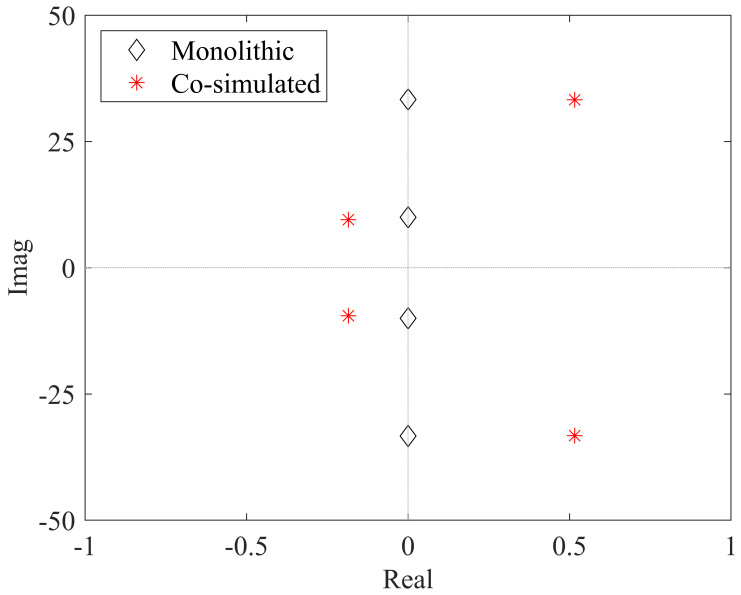


Figure 3: Primary poles of the monolithic and co-simulated undamped two-mass system.

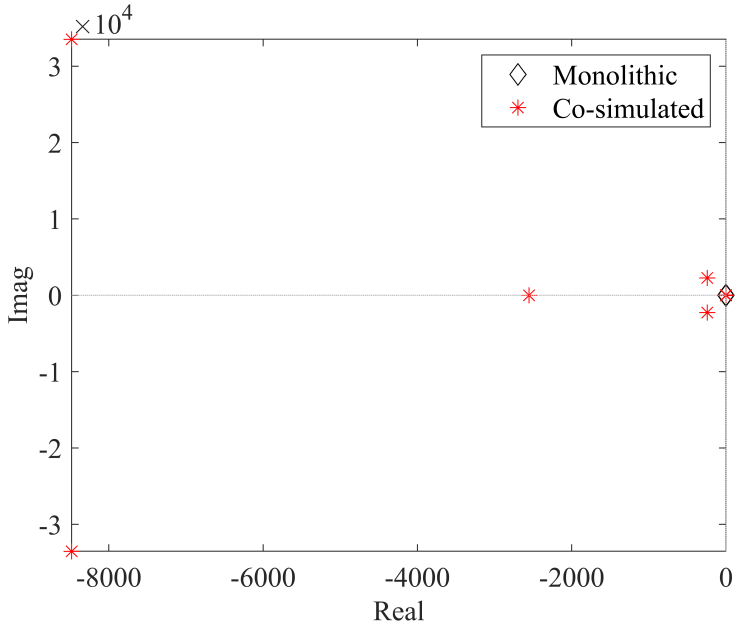


Figure 4: Latent roots of the co-simulated undamped two-mass system.

Table 1: Eigenstructure and modal parameters of the undamped two-mass system: monolithic and co-simulated.

Monolithic		**Co-simulated**	
$\lambda_{1,2}$	$\lambda_{3,4}$	$\tilde{\lambda}_{1,2}$	$\tilde{\lambda}_{3,4}$
$\pm 10.00j$	$\pm 33.32j$	$-0.19 \pm 9.51j$	$0.52 \pm 33.27j$
$\omega_{n_{1,2}}[\text{rad}/s]$	$\omega_{n_{3,4}}[\text{rad}/s]$	$\tilde{\omega}_{n_{1,2}}[\text{rad}/s]$	$\tilde{\omega}_{n_{3,4}}[\text{rad}/s]$
10.00	33.32	9.51	33.27
$\xi_{1,2}$	$\xi_{3,4}$	$\tilde{\xi}_{1,2}$	$\tilde{\xi}_{3,4}$
0	0	0.020	-0.015
$\mathbf{w}_{1,2}$	$\mathbf{w}_{3,4}$	$\tilde{\mathbf{w}}_{1,2}$	$\tilde{\mathbf{w}}_{3,4}$
$-0.9950j$ $-0.0995j$	$-0.0995j$ $0.9950j$	$-0.9950j$ $+0.0013 - 0.0994j$	$-0.0995j$ $-0.004 + 0.9950j$

Let us now consider the second scenario, i.e., the damped two-mass system. The co-simulated system with ZOH extrapolation and $T_s = 1$ ms leads to the poles summarized in Table 2 together with the related eigenvectors. The pole analysis highlights that, even though instability does not arise, the perturbation due to co-simulation alters the dynamics with respect to the monolithic system, as it will be assessed through the time-domain analysis in Section 5.2. Indeed, the co-simulated system features different damping ratios and natural frequencies because of the co-simulation interface. The same holds for the mode-shape perturbations, that change the spatial pattern of vibration.

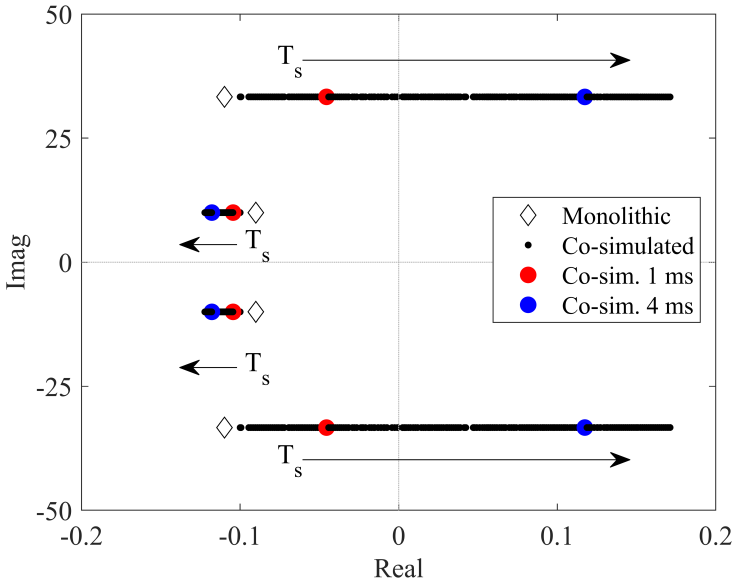


Figure 5: Primary poles of the damped two-mass system varying T_s.

Let us now increase T_s from 1 ms up to 4 ms. The eigenvalues of such a co-simulated system are listed in Table 2. The second pole pair $\tilde{\lambda}_{3,4}$ is particularly affected by the increase of T_s in the data exchange at the co-simulation interface. In particular its real part is the most affected, whose sign varies from negative to positive once T_s increases. It is evident that this co-simulated system, i.e., the one with $T_s = 4$ ms, features an unstable pole pair $\tilde{\lambda}_{3,4}$. Indeed, the phase lag introduced by $\mathbf{\Delta G_c}\,(s)$ is larger, and the phase and gain perturbations affect a broader range of frequencies. The monolithic system poles together with the co-simulated system poles varying the macro-step size T_s from 0 to 5 ms are shown in Fig. 5.

Table 2: Eigenstructure and modal parameters of the damped two-mass system: monolithic and co-simulated, $T_s = 1$ ms and $T_s = 4$ ms.

Monolithic		**Co-simulated** $T_s = 1$ ms		**Co-simulated** $T_s = 4$ ms	
$\lambda_{1,2}$	$\lambda_{3,4}$	$\tilde{\lambda}_{1,2}$	$\tilde{\lambda}_{3,4}$	$\tilde{\lambda}_{1,2}$	$\tilde{\lambda}_{3,4}$
$-0.09 \pm 10.00j$	$-0.11 \pm 33.32j$	$-0.11 \pm 10.00j$	$-0.05 \pm 33.32j$	$-0.11 \pm 10.00j$	$0.12 \pm 33.31j$
$\omega_{n_{1,2}}[\text{rad}/s]$	$\omega_{n_{3,4}}[\text{rad}/s]$	$\tilde{\omega}_{n_{1,2}}[\text{rad}/s]$	$\tilde{\omega}_{n_{3,4}}[\text{rad}/s]$	$\tilde{\omega}_{n_{1,2}}[\text{rad}/s]$	$\tilde{\omega}_{n_{3,4}}[\text{rad}/s]$
10.00	33.32	10.001	33.32	10.001	33.31
$\xi_{1,2}$	$\xi_{3,4}$	$\tilde{\xi}_{1,2}$	$\tilde{\xi}_{3,4}$	$\tilde{\xi}_{1,2}$	$\tilde{\xi}_{3,4}$
0.009	0.0033	0.011	0.0015	0.011	-0.0036
$\mathbf{w}_{1,2}$	$\mathbf{w}_{3,4}$	$\tilde{\mathbf{w}}_{1,2}$	$\tilde{\mathbf{w}}_{3,4}$	$\tilde{\mathbf{w}}_{1,2}$	$\tilde{\mathbf{w}}_{3,4}$
$-0.097 + 0.990j$ $-0.011 + 0.099j$	$0.0001 - 0.0995j$ $0.0312 + 0.9945j$	$-0.097 + 0.990j$ $-0.011 + 0.099j$	$0.0001 - 0.0995j$ $-0.0026 + 0.9945j$	$-0.097 + 0.990j$ $-0.015 + 0.098j$	$0.0001 - 0.0995j$ $-0.014 + 0.9950j$

3 Compensation through eigenstructure assignment

3.1 Objectives of the control

The perturbation introduced by $\mathbf{\Delta G_c}\,(s)$ should be compensated to enhance the accuracy of the co-simulation by making the monolithic and the co-simulated systems feature the same eigenstructure. Therefore, this paper here proposes a co-simulation compensation strategy based on the technique of eigenstructure assignment: to the best of the authors' knowledge, this approach has never been explored in the field of co-simulation. This method is sometimes adopted in the field of feedback control design [2, 3, 31, 46, 1, 7, 35, 4], although more attention is usually paid to just pole placement, while mode-shape assignment is often neglected due to larger difficulties in achieving the desired eigenvectors.

The co-simulation compensation is interpreted as a feedback control scheme, as shown in Fig. 6, that sketches the block-scheme representation in the Laplace domain of the co-simulated system together with the compensation $\mathbf{\Delta G_R}(s)$. The goal of $\mathbf{\Delta G_R}$ is to compute proper "virtual" forces $\mathbf{u}(s)$ to be distributed along the system through matrix $\mathbf{B}$, to correct the co-simulated model (described in Fig. 6 by its receptance matrix $\mathbf{H_c}(s) = (\mathbf{G_t}(s) + \mathbf{\Delta G_c}(s))^{-1}$). It should be noted that the proposed control scheme does not use any reference to track, thus treating control as a regulation problem.

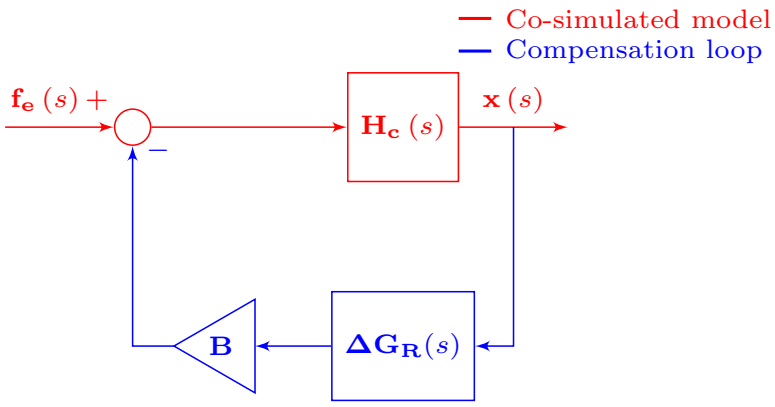


Figure 6: General block-scheme of the co-simulation model with the compensator.

An effective way to perform eigenstructure assignment is exploiting the displacement-velocity-acceleration feedback [36], i.e., the so-called state plus state-derivative feedback. Hence, $\mathbf{u}(s)$ is a linear function of $\mathbf{x}(s)$, $s\mathbf{x}(s)$, and $s^2\mathbf{x}(s)$ through the compensation gains $\mathbf{d}$, $\mathbf{f}$, $\mathbf{g}$, respectively the acceleration, velocity and displacement control gain matrices. In the field of vibration control with "perfect measurement" (i.e., feedback of the actual variables), $\mathbf{u}(s)$ would have been computed as $\mathbf{u}(s) = -\mathbf{R}(s)\mathbf{x}(s) = -\left(s^2\mathbf{d} + s\mathbf{f} + \mathbf{g}\right)\mathbf{x}(s)$, where $\mathbf{R}(s)$ is the feedback controller (often denoted as the compensator). In the case of co-simulation, some variables are replaced by those exchanged by the co-simulation manager and hence perturbed by $\mathrm{D}(s)$. Therefore, in a general form, the control law is:

$$\mathbf{u}(s) = -\mathbf{\Delta G_R}(s)\mathbf{x}(s) \tag{18}$$

where $\mathbf{\Delta G_R}(s)$ depends on the co-simulation compensator $\mathbf{R}(s)$ and $\mathrm{D}(s)$. The explicit structure of $\mathbf{\Delta G_R}(s)$ is not provided here since it depends on the system to be compensated as well as on the number of the compensation forces that will be adopted, hence it will be described in the following examples. Such a compensator has been chosen since it allows assigning the eigenstructure with low order controllers, i.e., without the introduction of several poles and ze-

ros due to the controller itself, as it might happen by using lead-lag transfer functions. If the full state and its derivative are not available to compute $\mathbf{u}(s)$ (i.e., some entries of $\mathbf{d}$, $\mathbf{f}$, $\mathbf{g}$ are forced to be zero), the compensator is said to be an output feedback controller, and some limitations on the achievable results are expected, as discussed in Section 4.1.

3.2 Compensator design

The dynamics of the compensated co-simulated system matches that of the monolithic one if the two systems have the same eigenstructure and the same static gain. It should be noted that matching the poles, i.e., natural frequencies and damping, is not enough since the compensation might lead to two isospectral systems with significantly different mode shapes. As far as the secondary roots are concerned, since they are not present in the monolithic model, these should simply lie in the left half complex plane, as discussed separately.

Let us analyze the dynamics of the compensated co-simulated system through the Laplace domain representation:

$$\left(\mathbf{G_t}(s) + \mathbf{\Delta G_c}(s)\right)\mathbf{x}(s) = \mathbf{f_e}(s) - \mathbf{B\Delta G_R}(s)\,\mathbf{x}(s) \tag{19}$$

Therefore the compensator $\mathbf{\Delta G_R}(s)$ should be designed by imposing that all the eigenpairs $(\lambda_i, \mathbf{w}_i)$ of the monolithic system must solve the eigenproblem of the co-simulated compensated one:

$$\left(\mathbf{G_t}(\lambda_i) + \mathbf{\Delta G_c}(\lambda_i) + \mathbf{B\Delta G_R}(\lambda_i)\right)\mathbf{w}_i = \mathbf{0}, \quad i = 1, ..., 2N \tag{20}$$

The eigenproblem of the monolithic system in Eq. (5) points out that $\mathbf{G_t}(\lambda_i)\,\mathbf{w}_i = \mathbf{0}$ for $i = 1, ..., 2N$. Hence Eq. (20) becomes:

$$\left(\mathbf{\Delta G_c}(\lambda_i) + \mathbf{B\Delta G_R}(\lambda_i)\right)\mathbf{w}_i = \mathbf{0}, \quad i = 1, ..., 2N \tag{21}$$

The eigenpairs $(\lambda_i, \mathbf{w}_i)$ to be assigned are obtained through the eigenstructure analysis of $\mathbf{G_t}(s)$. However, it is important to remark that it not necessary to know a-priori the monolithic system model, indeed the transfer function $\mathbf{G_t}(s)$ can be obtained by assembling the transfer functions of the subsystem and the one of the coupling as already shown through Eq. (11). Furthermore,

in this work, we assume that all subsystem internals are fully known for the sake of clarity. In practice, subsystems can behave as black boxes, and the compensation solution would require the use of subsystem characterization or identification, e.g., by means of approaches like the ones in [33] or [22].

Since the eigenpairs λ_i and $\mathbf{w}_i$ are imposed as the ones solving the eigenproblem of the monolithic system, $\mathbf{\Delta G_R}\,(s)$ becomes a linear function of the gains $\mathbf{d}$, $\mathbf{f}$ and $\mathbf{g}$. Hence, Eq. (21) is cast as a linear system whose unknowns are $\mathbf{d}$, $\mathbf{f}$, and $\mathbf{g}$, that is summarized in the following form:

$$\mathbf{L}\mathbf{k} = \mathbf{n} \tag{22}$$

where $\mathbf{L} \in \mathbb{C}^{(2N\cdot p)\times(m\cdot 3r)}$, $\mathbf{n} \in \mathbb{C}^{2N\cdot p}$, and $\mathbf{k} = \begin{Bmatrix} \mathbf{d} \\ \mathbf{f} \\ \mathbf{g} \end{Bmatrix} \in \mathbb{R}^{3mr}$. p, m and r are three variables here exploited to define the dimensions of the matrices and vectors involved in Eq. (22). Sizes p, m, and r depend on the co-simulated system under investigation and on the choice of the compensator. It follows that a clearer explanation on the typical topology of $\mathbf{L}$ and $\mathbf{n}$ is provided through the examples along the paper. However, some observations can be carried out already. The size p is not smaller than the number of non-zero rows of $\mathbf{\Delta G_c}$. The number of feedback variables is $3r$, where $r \leq N$ is the number of coordinates that are available in terms of both position, speed, and acceleration. The number of independent compensation forces is $m = \text{rank}\,(\mathbf{B})$, the so-called "rank of the control".

The terms r and m should be chosen to ensure that the number of gains of the compensator is greater than the number of equations, i.e., the linear system is underdetermined, to allow for the stabilization of the secondary roots [7]. Hence, $m \cdot 3r > 2N \cdot p$. For example, if full state plus derivative control is assumed, then $r = N$, hence it must hold that $3m > 2p$, i.e., matrix $\mathbf{B}$ should be chosen such that the number of independent actuation forces should be greater than the number of non-zero rows in $\mathbf{\Delta G_c}$, that are in practice the rows that perturb the eigenstructure of the monolithic system.

Under the assumption that $m \cdot 3r > 2N \cdot p$, the solution of Eq. (22) is formulated in the

general form [7]:

$$\mathbf{k} = \mathbf{k_0} + \mathbf{V}\mathbf{k_r} \tag{23}$$

The particular solution of the non-homogeneous Eq. (23) is $\mathbf{k_0}$, that can be computed through several methods (such as through the pseudoinverse or the QR decomposition). $\mathbf{V}\mathbf{k_r}$ is the solution of the homogeneous equation $\mathbf{L}\mathbf{V}\mathbf{k_r} = \mathbf{0}$. Matrix $\mathbf{V} \in \mathbb{R}^{3mr \times (3mr-2Np)}$ belongs to the null-space of $\mathbf{L}$, while $\mathbf{k_r} \in \mathbb{R}^{3mr-2Np}$ is an arbitrary vector which can be chosen to exploit the redundancy in the solution of Eq. (23). In this way, any choice of $\mathbf{k_r}$ does not cause spillover on the primary poles, and related eigenvectors, assigned through $\mathbf{k_0}$. In this paper $\mathbf{k_r}$ is exploited to place the latent roots due to time-delay in the left half of the complex plane, to make the compensated co-simulated system stable.

3.3 Compensator synthesis for the two-mass system

Let us consider the model of the two-mass system analyzed in Section 2.3. The compensation consists of two independent forces computed through position, speed, and acceleration feedback of both the coordinates, defined in the Laplace domain as follows:

$$\begin{cases} \mathrm{u}_1(s) = -\left(\left(s^2 d_1 + s f_1 + g_1\right)\mathrm{x}_1(s) + \left(s^2 d_2 + s f_2 + g_2\right)\mathrm{x}_2^*(s)\right) \\ \mathrm{u}_2(s) = -\left(\left(s^2 d_3 + s f_3 + g_3\right)\mathrm{x}_1^*(s) + \left(s^2 d_4 + s f_4 + g_4\right)\mathrm{x}_2(s)\right) \end{cases} \tag{24}$$

where d_i, f_i, and g_i, with $i = 1, ..., 4$ are respectively the i-th acceleration, velocity, and displacement compensation gain. Hence, they are the entries of vectors $\mathbf{d}$, $\mathbf{f}$, and $\mathbf{g}$. Further, $\mathrm{x}_1^*(s) = \mathrm{D}(s)\,\mathrm{x}_1$ and $\mathrm{x}_2^*(s) = \mathrm{D}(s)\,\mathrm{x}_2$, as shown in Eq. (12). Eq. (24) can be written with the general notation employed in Eq. (18) as:

$$\begin{Bmatrix} \mathrm{u}_1(s) \\ \mathrm{u}_2(s) \end{Bmatrix} = -\begin{bmatrix} s^2 d_1 + s f_1 + g_1 & \left(s^2 d_2 + s f_2 + g_2\right)\mathrm{D}(s) \\ \left(s^2 d_3 + s f_3 + g_3\right)\mathrm{D}(s) & s^2 d_4 + s f_4 + g_4 \end{bmatrix} \begin{Bmatrix} \mathrm{x}_1(s) \\ \mathrm{x}_2(s) \end{Bmatrix} \tag{25}$$

with $\mathbf{B} = \mathbf{I}_2$, i.e., the size-two identity matrix. The eigenproblem in Eq. (21) can be formulated using the following matrices:

$$\begin{aligned}
\mathbf{\Delta G_c}\left(\lambda_i\right) &= \begin{bmatrix} 0 & \left(\lambda_i c_c + k_c\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right) \\ \left(\lambda_i c_c + k_c\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right) & \left(\lambda_i c_c + k_c\right)\left(D^2\left(\lambda_i\right) - 1\right) \end{bmatrix} \\
\mathbf{\Delta G_R}\left(\lambda_i\right) &= \begin{bmatrix} \left(\lambda_i\right)^2 d_1 + \lambda_i f_1 + g_1 & \left(\left(\lambda_i\right)^2 d_2 + \lambda_i f_2 + g_2\right)\mathrm{D}\left(\lambda_i\right) \\ \left(\left(\lambda_i\right)^2 d_3 + \lambda_i f_3 + g_3\right)\mathrm{D}\left(\lambda_i\right) & \left(\lambda_i\right)^2 d_4 + \lambda_i f_4 + g_4 \end{bmatrix}
\end{aligned} \tag{26}$$

Finally, the substitution of Eq. (26) into Eq. (21), yields the linear system that represents the eigenstructure assignment problem to be solved for co-simulation compensation:

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{L_{a,i}} & \mathbf{L_{v,i}} & \mathbf{L_{d,i}} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{Bmatrix} \mathbf{d} \\ \mathbf{f} \\ \mathbf{g} \end{Bmatrix} = \begin{Bmatrix} n_{1,i} \\ n_{2,i} \end{Bmatrix}, \quad i = 1, ..., 2N \tag{27}$$

with:

$$\begin{aligned}
\mathbf{L_{a,i}} &= \begin{bmatrix} \lambda_i^2 \mathbf{w}_i^{(1)} & \lambda_i^2 \mathbf{w}_i^{(2)} \mathrm{D}\left(\lambda_i\right) & 0 & 0 \\ 0 & 0 & \lambda_i^2 \mathbf{w}_i^{(1)} \mathrm{D}\left(\lambda_i\right) & \lambda_i^2 \mathbf{w}_i^{(2)} \end{bmatrix} \\
\mathbf{L_{v,i}} &= \begin{bmatrix} \lambda_i \mathbf{w}_i^{(1)} & \lambda_i \mathbf{w}_i^{(2)} \mathrm{D}\left(\lambda_i\right) & 0 & 0 \\ 0 & 0 & \lambda_i \mathbf{w}_i^{(1)} \mathrm{D}\left(\lambda_i\right) & \lambda_i \mathbf{w}_i^{(2)} \end{bmatrix} \\
\mathbf{L_{d,i}} &= \begin{bmatrix} \mathbf{w}_i^{(1)} & \mathbf{w}_i^{(2)} \mathrm{D}\left(\lambda_i\right) & 0 & 0 \\ 0 & 0 & \mathbf{w}_i^{(1)} \mathrm{D}\left(\lambda_i\right) & \mathbf{w}_i^{(2)} \end{bmatrix} \\
\begin{Bmatrix} n_{1,i} \\ n_{2,i} \end{Bmatrix} &= - \begin{Bmatrix} \left(\lambda_i c_c + k_c\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right) \mathbf{w}_i^{(2)} \\ \left(\lambda_i c_c + k_c\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right) \mathbf{w}_i^{(1)} + \left(\lambda_i c_c + k_c\right)\left(D^2\left(\lambda_i\right) - 1\right) \mathbf{w}_i^{(2)} \end{Bmatrix} \\
\mathbf{d} &= \begin{Bmatrix} d_1 & \dots & d_4 \end{Bmatrix}^T, \quad \mathbf{f} = \begin{Bmatrix} f_1 & \dots & f_4 \end{Bmatrix}^T, \quad \mathbf{g} = \begin{Bmatrix} g_1 & \dots & g_4 \end{Bmatrix}^T
\end{aligned} \tag{28}$$

where $\mathbf{w}_i^{(j)}$ denotes the j-th entry of $\mathbf{w}_i$.

The solution of the linear system in Eq. (27) is obtained as follows: the particular solution $\mathbf{k_0}$ has been computed through the MATLAB function $mldivide$, by recasting the linear system in Eq. (22) into a numerically robust form [7, 35]:

$$\begin{bmatrix} \text{Real}\left(\mathbf{L}\right) \\ \text{Imag}\left(\mathbf{L}\right) \end{bmatrix} \mathbf{k_0} = \begin{bmatrix} \text{Real}\left(\mathbf{n}\right) \\ \text{Imag}\left(\mathbf{n}\right) \end{bmatrix} \tag{29}$$

Indeed, the obtained compensation gains should be real to make feasible the controller implementation in the time-domain [34]. Matrix $\mathbf{V}$ has been computed through the MATLAB function $null$ and vector $\mathbf{k_r}$ has been obtained through random search algorithms, until a solution that stabilizes the secondary roots has been found.

All the equations proposed so far hold for both the damped and undamped systems. Let us first consider the undamped two-mass system (scenario 1). The solution of the linear system in Eq. (27) has led to the following control gains:

- $\mathbf{d} = \left\{ -0.0035 \quad 0.0004 \quad -0.0108 \quad -0.0012 \right\}^T$;
- $\mathbf{f} = \left\{ 0.0000 \quad -0.0498 \quad -0.0499 \quad 0.0994 \right\}^T$;
- $\mathbf{g} = \left\{ -0.4222 \quad 0.7846 \quad -1.0729 \quad -0.1962 \right\}^T$.

The compensated co-simulated system features the $2N$ eigenpairs $\left(\bar{\lambda}_i, \bar{\mathbf{w}}_i\right)$ listed in Table 3: the correct assignment of the eigenstructure is clearly obtained since the N pairs $\left(\bar{\lambda}_i, \bar{\mathbf{w}}_i\right)$ match those of the theoretical, monolithic model $\left(\lambda_i, \mathbf{w}_i\right), i = 1, ..., 2N$. This result is corroborated by the analysis of the eigenloci in Fig. 7, that highlights the primary poles. The latent roots due to the exponential terms of the co-simulation manager are shown in Fig. 8, that clearly shows that the compensated system is stable since all the roots are clustered in the left half of the complex plane. Table 3 collects also the modal parameters (natural frequencies and damping ratios) and the percentage error of the i-th natural frequency of the co-simulated system with respect to those of the monolithic one denoted through e_{ω_i}. Further, it lists also the difference between the i-th modal damping of the monolithic system with respect to the co-simulated system denoted by e_{ξ_i}. The effectiveness of the proposed compensation approach is corroborated by these two indexes whose values approach zero, i.e., the eigenstructure of the co-simulated system with compensation matches the eigenstructure of the monolithic system.

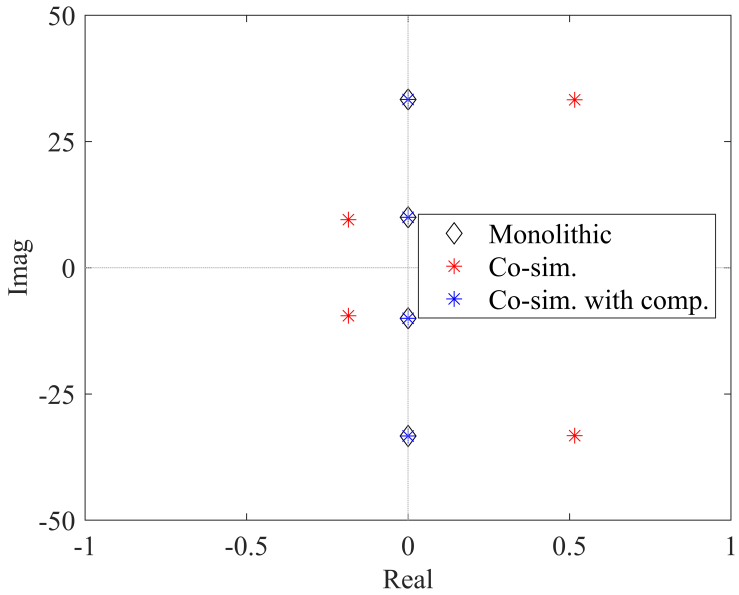


Figure 7: Eigenloci of the undamped two-mass system: monolithic and co-simulated without and with compensation.

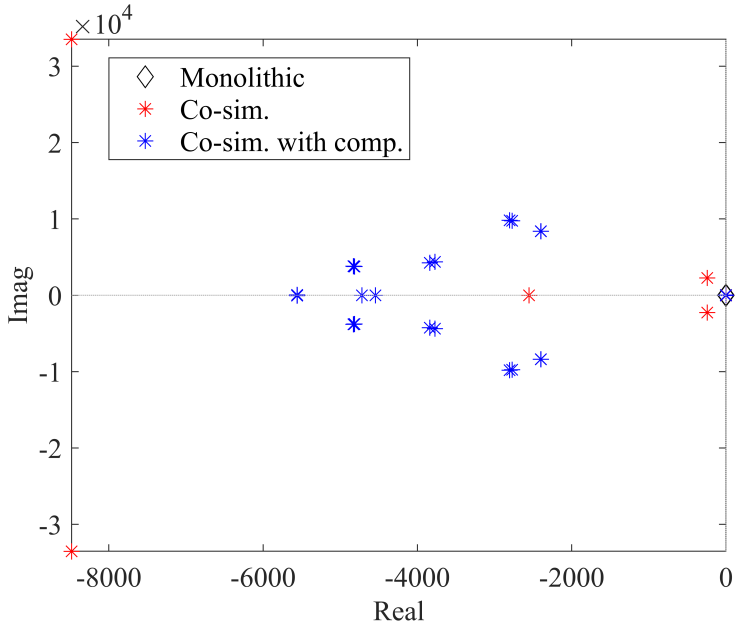


Figure 8: Latent roots of the undamped two-mass system: monolithic and co-simulated without and with compensation.

Table 3: Eigenstructure and modal parameters of the undamped two-mass system: monolithic and co-simulated without and with compensation.

Monolithic		**Co-simulated**		**Co-simulated compensated**	
$\lambda_{1,2}$	$\lambda_{3,4}$	$\tilde{\lambda}_{1,2}$	$\tilde{\lambda}_{3,4}$	$\bar{\lambda}_{1,2}$	$\bar{\lambda}_{3,4}$
$\pm 10.00j$	$\pm 33.32j$	$-0.19 \pm 9.51j$	$0.52 \pm 33.27j$	$\pm 10.00j$	$\pm 33.32j$
$\mathbf{w}_{1,2}$	$\mathbf{w}_{3,4}$	$\tilde{\mathbf{w}}_{1,2}$	$\tilde{\mathbf{w}}_{3,4}$	$\bar{\mathbf{w}}_{1,2}$	$\bar{\mathbf{w}}_{3,4}$
-0.9950 -0.0995	-0.0995 0.9950	-0.9951 $-0.09948 \mp 0.0013j$	$-0.9951 \pm 0.0004j$ 0.9951	-0.9950 -0.0995	-0.0995 0.9950
$\omega_{n_{1,2}}[\text{rad}/s]$	$\omega_{n_{3,4}}[\text{rad}/s]$	$\tilde{\omega}_{n_{1,2}}[\text{rad}/s]$	$\tilde{\omega}_{n_{3,4}}[\text{rad}/s]$	$\bar{\omega}_{n_{1,2}}[\text{rad}/s]$	$\bar{\omega}_{n_{3,4}}[\text{rad}/s]$
10.00	33.32	9.51	33.27	10.00	33.32
$e_{\omega_{1,2}}[\%]$	$e_{\omega_{3,4}}[\%]$	$e_{\omega_{1,2}}[\%]$	$e_{\omega_{3,4}}[\%]$	$e_{\omega_{1,2}}[\%]$	$e_{\omega_{3,4}}[\%]$
–	–	-4.88	-0.14	0.00	0.00
$\xi_{1,2}$	$\xi_{3,4}$	$\tilde{\xi}_{1,2}$	$\tilde{\xi}_{3,4}$	$\bar{\xi}_{1,2}$	$\bar{\xi}_{3,4}$
0	0	0.020	-0.015	0	0
$e_{\xi_{1,2}}$	$e_{\xi_{3,4}}$	$e_{\xi_{1,2}}$	$e_{\xi_{3,4}}$	$e_{\xi_{1,2}}$	$e_{\xi_{3,4}}$
–	–	0.020	-0.015	0.00	0.00

Let us now consider the damped two-mass system (scenario 2). In this case, two values of macro-step size are considered $T_s = 1$ ms and $T_s = 4$ ms, to discuss the effect of the step size. The dominant poles of the monolithic and the co-simulated compensated systems are listed in Table 4 together with the compensator gains. It is evident that the compensated system dynamics, i.e., natural frequencies, modal damping, and eigenvectors, fulfills the specifications. This result is corroborated by the pole map provided in Fig. 9, indeed the poles spillover due to co-simulation vanishes. The analysis of the latent roots due to time-delay, here omitted for brevity, highlights that those lie in the left half of the complex plane, hence the compensated system is stable, as it will be shown through the time domain simulations proposed in Sections 5.2.2 and 5.2.3.

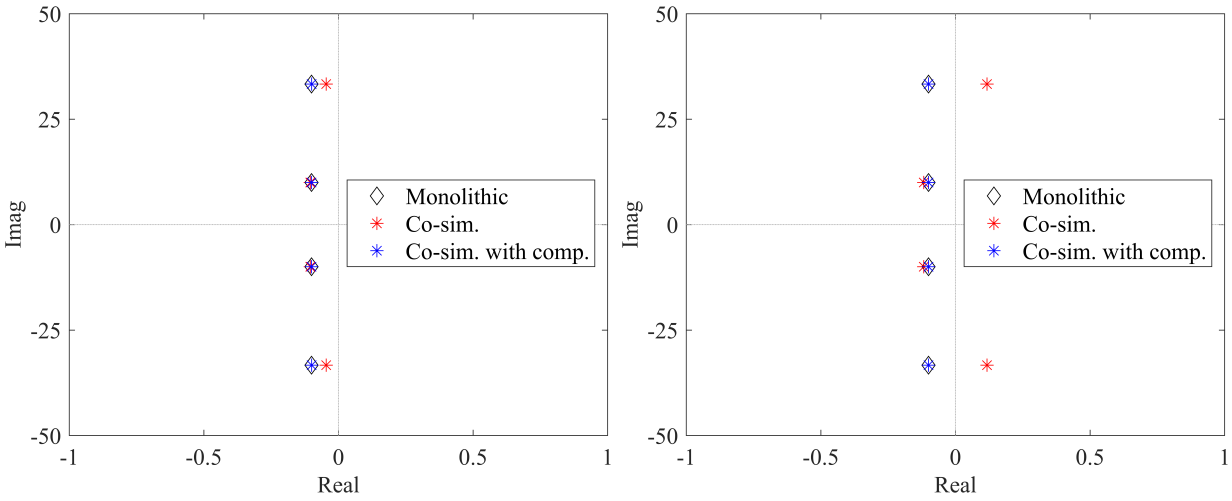
4 Controllability and eigenstructure assignability

4.1 General discussion and theoretical concepts

Two cruxes of eigenstructure assignment through active control, that should be carefully addressed in co-simulation compensation too, are controllability and eigenstructure assignability.

Table 4: Compensator gains and dominant poles of the damped two-mass system: monolithic and co-simulated with compensator, $T_s = 1$ ms and $T_s = 4$ ms.

$\mathbf{T_s}$	**Monolithic**		**Co-simulated compensated**	
	$\lambda_{1,2}$	$\lambda_{3,4}$	$\bar{\lambda}_{1,2}$	$\bar{\lambda}_{3,4}$
1 ms	$-0.1 \pm 10.00j$	$-0.1 \pm 33.32j$	$-0.1 \pm 10.00j$	$-0.1 \pm 33.32j$
$\mathbf{d} = \left\{-0.0036 \quad 0.0003 \quad 0.0018 \quad 0.0001\right\}^T$				
$\mathbf{f} = \left\{-0.0007 \quad -0.0494 \quad -0.0497 \quad 0.100\right\}^T$				
$\mathbf{g} = \left\{-0.4357 \quad 0.7596 \quad 0.1896 \quad -0.0743\right\}^T$				
4 ms	$-0.1 \pm 10.00j$	$-0.1 \pm 33.32j$	$-0.1 \pm 10.00j$	$-0.1 \pm 33.32j$
$\mathbf{d} = \left\{0.0042 \quad -0.0003 \quad 0.0009 \quad -0.0004\right\}^T$				
$\mathbf{f} = \left\{0.0008 \quad -0.2011 \quad -0.1997 \quad 0.3994\right\}^T$				
$\mathbf{g} = \left\{0.4564 \quad -0.3928 \quad 0.1230 \quad 0.0528\right\}^T$				



(a) $T_s = 1$ ms (b) $T_s = 4$ ms

Figure 9: Primary poles of the monolithic and co-simulated damped two-mass system without and with compensation: (a) $T_s = 1$ ms, (b) $T_s = 4$ ms.

In this Section, a brief general discussion is provided by recalling some meaningful results with reference to the typical state-feedback control architecture for the linear system in Eq. (3), that leads to the following equations governing the closed-loop system (without loss of generality, it is set $\boldsymbol{f}_e(t) = \mathbf{0}$):

$$\mathbf{M}\ddot{\boldsymbol{x}}(t) + \mathbf{C}\dot{\boldsymbol{x}}(t) + \mathbf{K}\boldsymbol{x}(t) = -\mathbf{B}\left(\mathbf{f_s}^T\dot{\boldsymbol{x}}(t) + \mathbf{g_s}^T\boldsymbol{x}(t)\right) \tag{30}$$

where $\mathbf{f_s}$ and $\mathbf{g_s}$ are respectively the state-feedback velocity and displacement gain vectors.

A linear system is said to be controllable if, for any initial state, there exists a sequence of inputs that leads to any arbitrarily chosen final state [17]. This is equivalent to the property that the poles of the controlled system can be arbitrarily assigned through state feedback, i.e., through displacement and speed feedback as in Eq. (30). Acceleration feedback is here omitted for brevity, however state-derivative feedback has been proved to be equivalent to state-feedback (see e.g., [5]). An effective way to assess controllability for second order systems is through the following condition [15]:

$$\text{rank}\left(\begin{bmatrix}\lambda_i^2\mathbf{M}+\lambda_i\mathbf{C}+\mathbf{K} & \mathbf{B}\end{bmatrix}\right) = N, \quad \forall\lambda_i \tag{31}$$

Equation (31) shows that controllability depends on the number and on the placement of the control forces. The wide literature of control shows, however, that usually "few" forces suffice to ensure controllability, i.e., $m \ll N$ often allows assigning all the poles, provided that a wise placement of the actuators is done. This is typical in underactuated systems, where rank-one control is widely adopted [31, 37, 35, 4].

It should be noted that the assumption of displacement and speed feedback of all the coordinates is of fundamental importance: removing some of the feedback variables, i.e., performing output feedback, severely affects the possibility of arbitrarily assigning the poles.

As far as eigenstructure assignability is concerned, controllability is not enough [2]. Hence, a more severe condition is to be satisfied to ensure that the arbitrary eigenpair $(\lambda_i, \mathbf{w}_i)$ can be assigned. Let us introduce the receptance matrix of the open loop system $\mathbf{H}(\lambda_i) = \left(\lambda_i^2\mathbf{M}+\lambda_i\mathbf{C}+\mathbf{K}\right)^{-1}$; an arbitrary eigenpair is assignable if and only if [37]:

$$\mathbf{w}_i \in \text{span}\left(\mathbf{H}(\lambda_i)\,\mathbf{B}\right) \tag{32}$$

Such a condition reveals that the i-th eigenvector $\mathbf{w}_i$ related to the eigenvalue λ_i is assignable if and only if it belongs to the subspace spanned by the columns of $\mathbf{H}(\lambda_i)\,\mathbf{B}$. Eq. (32) shows, again, that the choice of $\mathbf{B}$ is crucial. However, compared to controllability, the rank of $\mathbf{B}$ has a drastic effect on the assignability since $\text{span}\left(\mathbf{H}(\lambda_i)\,\mathbf{B}\right) \in \mathbb{C}^{N\times m}$. Hence, the number of entries of each eigenvector that can be exactly assigned is m. For example, if $m = 1$, there is no possibility of assigning the eigenvector. If the complete eigenstructure should be assigned through state-feedback control, N control forces are therefore required.

Again, as in the discussion of controllability, if output feedback is used instead of state feedback, the possibilities of eigenvector assignment are further reduced.

4.2 On the eigenstructure assignability in co-simulation

The conditions provided in the previous Section are useful in the case of state-feedback control of the monolithic system, and provide a general explanation of these two concepts. However, in the ad-hoc framework developed in this paper for the co-simulation compensation, the application of both conditions is not straightforward. On the other hand, eigenstructure assignability cannot be assessed through the monolithic model, since the perturbation due to $\mathbf{B}\boldsymbol{\Delta}\mathbf{G}_{\mathbf{R}}(s)$ should be accounted for as well.

An effective approach to assess if the primary eigenvalues and the related eigenvectors are assignable through the chosen correction forces is based on the analysis of Eq. (22). Indeed, the linear system $\mathbf{L}\mathbf{k} = \mathbf{n}$ has a solution if and only if:

$$\operatorname{rank}(\mathbf{L}) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{L} & \mathbf{n}\end{bmatrix}\right) \tag{33}$$

as stated by the well known Rouché-Capelli theorem.

Hence, the eigenstructure assignment problem in the presence of the chosen compensation, formulated in Eq. (22), admits a solution if and only if vector $\mathbf{n}$ belongs to the subspace spanned by the columns of matrix $\mathbf{L}$. In practice, such a condition summarizes the assessment of both controllability and eigenvector assignability.

It is worth noticing that $\mathbf{L}$ depends upon $\mathbf{B}$, $\mathbf{R}(s)$, and $\mathrm{D}(s)$ and on the variables assumed for feedback in the case of output feedback, as it will be shown for the two-mass system in Section 3.3. Hence, both the choice of the distribution of the forces along the system through $\mathbf{B}$, and the choice of the feedback signals to compute $\mathbf{u}(s)$ play a fundamental role in the problem solvability.

Equation (33) is therefore a relevant tool to understand which is the right number of independent forces (the so-called control rank) and their proper placement in the system. Additionally, if the full state is not available, it allows one to understand if more variables should be exchanged by the manager to allow the compensator exactly assigning the eigenstructure.

Finally, it should be pointed out that Eq. (33) does not predict whether or not the secondary roots can be stabilized. The stabilization of such roots exploits the redundancy of the linear system, and the higher the degree of redundancy, i.e., the size of $\mathbf{k_r}$, the more likely stabilization is.

4.3 Eigenstructure assignability in the two-mass system

This Section investigates eigenstructure assignability through the topology of $\mathbf{B\Delta G_R}\left(s\right)$ chosen in Section 3.3 for the two-mass system.

Let us first assume that two independent compensation forces are applied through the force distribution matrix $\mathbf{B} = \mathbf{I}_2$, and those feedback variables include position, speed and acceleration feedback of all the coordinates (x_1, x_2^*, x_1^* and x_2), as defined in Eq. (24). The application of the assignability test in Eq. (33) yields: $\text{rank}\left(\mathbf{L}\right) = \text{rank}\left(\begin{bmatrix}\mathbf{L} & \mathbf{n}\end{bmatrix}\right) = 8$. This result holds for both the undamped and damped systems (the latter ones have been investigated for both $T_s = 1$ ms and $T_s = 4$ ms). Hence, the design of $\mathbf{B\Delta G_R}\left(s\right)$ as proposed in Section 3.3 is adequate to compensate co-simulation through the proposed eigenstructure assignment technique.

Let us now investigate the effect of removing one of the signals fed back to the compensator, which leads to an output-feedback control scheme. By means of example, let us omit x_2^*. In this case, and for all the three systems previously investigated, the assignability condition is not fulfilled since $\text{rank}\left(\mathbf{L}\right) = 7 \neq \text{rank}\left(\begin{bmatrix}\mathbf{L} & \mathbf{n}\end{bmatrix}\right) = 8$. It can be shown that in the general case of a damped two-mass system, to ensure eigenstructure assignability, it is not possible to neglect any of the above mentioned signals to tune the compensator.

Let us now assume that only one force is employed to compensate the co-simulation. This is the so-called "rank-one" control, i.e., $\text{rank}\left(\mathbf{B}\right) = 1$, and we assume $\mathbf{B} = \begin{Bmatrix}1 & -\mathrm{D}\left(s\right)\end{Bmatrix}^T$, meaning that one compensation force is applied to both masses with opposite signs. Clearly, the compensation term acting on the second mass is multiplied by $\mathrm{D}\left(s\right)$ since, once the control force is computed in subsystem 1, it must be communicated to subsystem 2 to be applied, i.e., it behaves as a coupling variable. In this scenario the compensation forces in Eq. (24) become:

$$\mathrm{u}\left(s\right) = -\left(\left(s^2 d_1 + s f_1 + g_1\right)\mathrm{x}_1\left(s\right) + \left(s^2 d_2 + s f_2 + g_2\right)\mathrm{x}_2^*\left(s\right)\right) \tag{34}$$

Hence $\mathbf{B\Delta G_R}(s)$ becomes:

$$\mathbf{B\Delta G_R}(s) = \begin{bmatrix} s^2 d_1 + s f_1 + g_1 & \left(s^2 d_2 + s f_2 + g_2\right) \mathrm{D}(s) \\ -\left(s^2 d_1 + s f_1 + g_1\right) \mathrm{D}(s) & -\left(s^2 d_2 + s f_2 + g_2\right) D^2(s) \end{bmatrix} \tag{35}$$

Obviously $\mathbf{\Delta G_c}(s)$, describing the perturbation due to co-simulation, is not affected by the variation of the compensator architecture.

Once the compensation eigenproblem $\left(\mathbf{\Delta G_c}(\lambda_i) + \mathbf{B\Delta G_R}(\lambda_i)\right)\mathbf{w}_i = \mathbf{0}$ is written for the $2N$ theoretical eigenpairs to be assigned, it is possible to formulate the linear system for the eigenstructure assignment with the canonical form $\mathbf{Lk} = \mathbf{n}$ adopted in this paper, where $\mathbf{k}$ collects in this case the 6 unknown control gains. The mathematical manipulations are here omitted for brevity.

The application of the Rouché-Capelli theorem in the case of rank-one control gives: $\mathrm{rank}(\mathbf{L}) = 4 \neq \mathrm{rank}\left(\begin{bmatrix}\mathbf{L} & \mathbf{n}\end{bmatrix}\right) = 5$ for the three systems under investigation. Hence, rank-one control is not adequate to perform eigenstructure assignment.

To briefly summarize the analysis proposed in this Section, it is proved that co-simulation compensation through eigenstructure assignment for the two-mass system under investigation is achievable through the compensator architecture proposed in Section 3.3. In contrast, neglecting compensation forces or feedback signals is not possible as evidenced by the eigenstructure assignability condition.

5 Co-simulation examples: the two-mass system

5.1 Co-simulator implementation details

A software code has been developed in MATLAB to carry out the co-simulation of the linear systems used as benchmarks in this paper. This co-simulation framework consists of a co-simulation manager to orchestrate the data exchange and synchronize the subsystem integrations following an explicit Jacobi scheme. The definitions of the subsystems have been particularized for each example, complying with the interface definition imposed by the co-simulation manager.

In general co-simulation implementations, the subsystems are often treated as black boxes,

in which only a reduced set of coupling variables are exposed. This assumption simplifies the definition of co-simulation standards, such as FMI (Functional Mockup Interface) [12]. However, for academic purposes, in this case the co-simulation manager has full access to the subsystem definitions, e.g., system parameters, states, extrapolation method, and internal step-sizes. The evaluation of the eigenstructure is performed once the system parameters are defined and the co-simulation manager is created, and therefore the model of the co-simulated system is known. The compensation scheme, as well as the compensator gains, are computed automatically before starting the co-simulation.

The compensator is implemented in the MATLAB co-simulation discrete-time software as sketched in Fig. 10 (where continuous-time notation is kept for uniformity with the paper text), with $\mathrm{f_{e,i}}\,(s)$ denoting the i-th external force, and where the following transfer functions are defined:

$$\begin{aligned} \mathrm{h}_1\,(s) &= \frac{1}{s^2 m_1 + s\,(c_1 + c_c) + (k_1 + k_c)} \\ \mathrm{h}_2\,(s) &= \frac{1}{s^2 m_2 + s c_2 + k_2} \\ \mathrm{g_c}\,(s) &= s c_c + k_c \end{aligned} \tag{36}$$

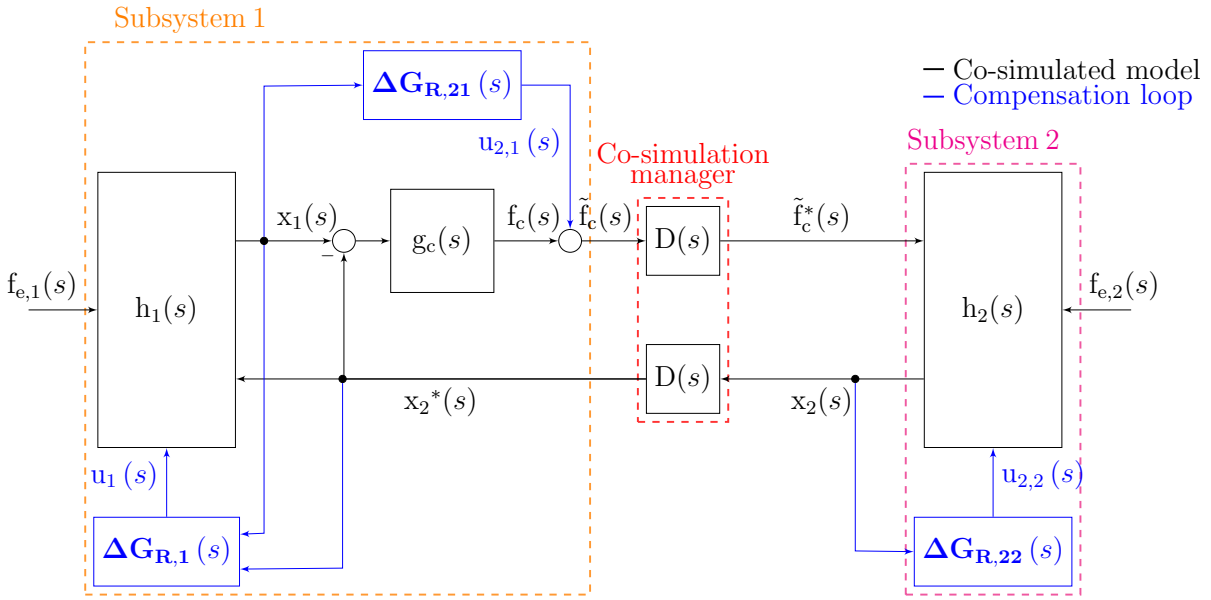


Figure 10: Implementation of the co-simulation model with the compensator for the two-mass system.

Since $\mathrm{x_1}^*\,(s)$ is not fed to subsystem 2 by the co-simulation manager, the compensation force $\mathrm{u_2}\,(s)$ applied to subsystem 2, as defined in Eq. (24), is split into two contributions to allow its

implementation:

$$\mathrm{u}_2(s) = \mathrm{u}_{2,1}{}^*(s) + \mathrm{u}_{2,2}(s) \tag{37}$$

$\mathrm{u}_{2,1}{}^*(s)$ is the non-collocated term, i.e., the contribution of $\mathrm{x}_1(s)$:

$$\mathrm{u}_{2,1}{}^*(s) = -\left(s^2 d_3 + s f_3 + g_3\right) \mathrm{x}_1(s)\, \mathrm{D}(s) \tag{38}$$

$\mathrm{u}_{2,2}(s)$ is the collocated one, i.e., the contribution of $\mathrm{x}_2(s)$:

$$\mathrm{u}_{2,2}(s) = -\left(s^2 d_4 + s f_4 + g_4\right) \mathrm{x}_2(s) \tag{39}$$

In turn, $\mathrm{u}_{2,1}{}^*$ can be written as the product between $\mathrm{D}(s)$ and the compensation force computed through $\mathrm{x}_1(s)$, referred to as $\mathrm{u}_{2,1}$:

$$\mathrm{u}_{2,1}(s) = -\left(s^2 d_3 + s f_3 + g_3\right) \mathrm{x}_1(s) \tag{40}$$

By adding $\mathrm{u}_{2,1}(s)$ to $\mathrm{f}_\mathrm{c}(s)$, the corrected coupling force $\tilde{\mathrm{f}_c}(s)$ is computed in Subsystem 1 and communicated to Subsystem 2 through the co-simulation manager, leading to $\tilde{\mathrm{f}_c}{}^*(s)$. $\mathrm{u}_{2,2}(s)$ is, instead, computed in Subsystem 2 and treated as an external force.

5.2 Co-simulations: free evolution

This Section compares the time-domain results of the simulations of the reference solution, the monolithic model, the co-simulated model without any compensation (i.e., in an "open loop" simulation), and the compensated co-simulated model (i.e., with the "closed loop" correction). These four scenarios will be henceforth denoted as "ref", "mono", "O.L." and "C.L." respectively. The reference solution is computed as [17]:

$$\boldsymbol{z}(t) = e^{\mathbf{A}t}\boldsymbol{z_0} \tag{41}$$

where the state-space representation is adopted, with $\mathbf{A}$ the state-matrix, $\boldsymbol{z}(t) = \left\{\boldsymbol{x}(t) \quad \dot{\boldsymbol{x}}(t)\right\}^T$ the state vector and $\boldsymbol{z_0} = \boldsymbol{z}(0) = \left\{\boldsymbol{x_0} \quad \dot{\boldsymbol{x}}_\mathbf{0}\right\}^T$ the state initial condition.

The free evolution from the sample initial conditions $\boldsymbol{x_0} = \begin{Bmatrix} 0 & 0 \end{Bmatrix}^T$ m and $\dot{\boldsymbol{x}}_0 = \begin{Bmatrix} 100 & -100 \end{Bmatrix}^T$ m/s, proposed in [42] and adopted also in [20, 38], is simulated for the three different scenarios already discussed along the paper

- Undamped two-mass system with $T_s = 1$ ms (Section 5.2.1);
- Damped two-mass system with $T_s = 1$ ms (Section 5.2.2);
- Damped two-mass system with $T_s = 4$ ms (Section 5.2.3);

The system parameters and the compensation gain vectors for the three scenarios are those introduced in Sections 2.3.2 and 3.3.

The symplectic Euler formula was used as numerical integrator inside both subsystems in the co-simulation and also in the evaluation of the monolithic solution. The microstep size for each subsystem is set equal to T_s. In all the scenarios, ZOH extrapolation is assumed. The normalized root mean square (NRMS) error of a signal $x(t)$, with respect to its reference solution $x_{\text{ref}}(t)$, is employed to evaluate the accuracy of the simulation results [14]:

$$e^x_{NRMS} = \frac{\sqrt{\frac{\sum_{t=0}^{T} \left(x(t) - x_{\text{ref}}(t)\right)^2}{T}}}{x_{\text{ref}}^{\max} - x_{\text{ref}}^{\min}} \tag{42}$$

The maximum and minimum value of $x_{\text{ref}}(t)$ over the simulation time T, denoted $x_{\text{ref}}^{\max}$ and $x_{\text{ref}}^{\min}$ respectively, are assumed for normalization.

5.2.1 Free evolution: undamped two-mass system

The time-domain simulations in Figs. 11 and 12 highlight that the co-simulated undamped system without compensation is unstable, i.e., leads to diverging values of the coordinates. This result confirms the analysis of the eigenstructure, as proposed in Section 2.3.2, that yields a pole pair in the right half of the complex plane.

The application of the proposed co-simulation compensation scheme stabilizes the co-simulation and ensures accurate estimates that almost match the one provided by the reference solution and by the monolithic model, as expected by the matching of the eigenstructures. This result

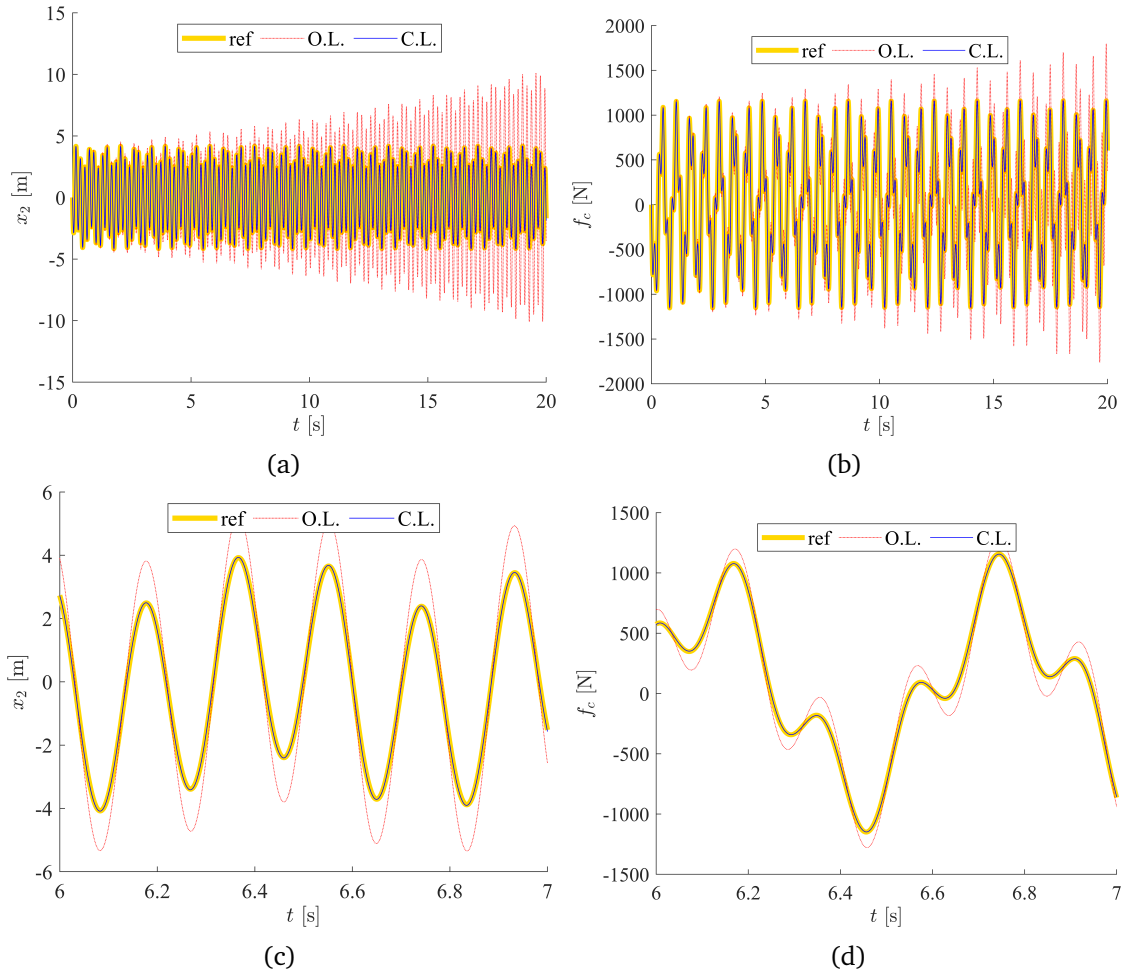


Figure 11: Undamped two-mass system time-domain free evolution co-simulation with and without compensation: (a) x_2, (b) f_c, (c) x_2 magnified view, (d) f_c magnified view.

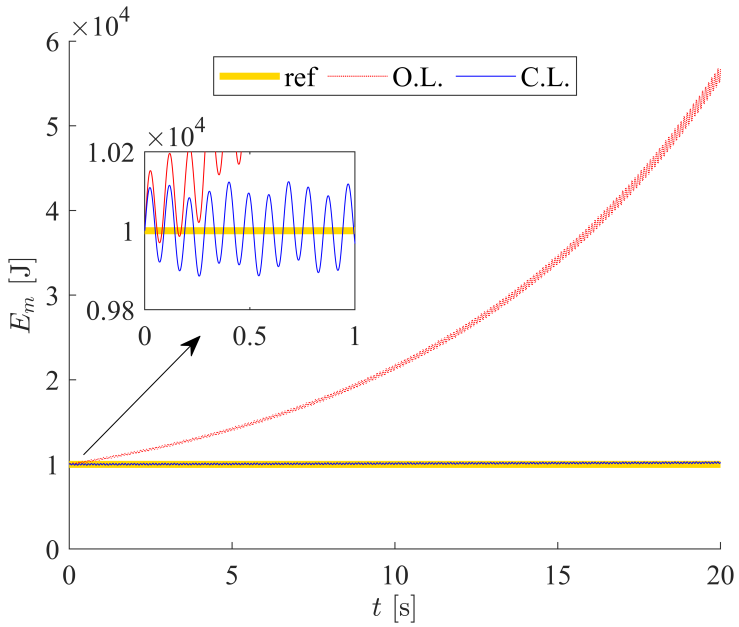


Figure 12: Undamped two-mass system time-domain free evolution co-simulation with and without compensation: mechanical energy.

is clearly shown in Figs. 11 and 12, and through the NRMS errors listed in Table 5. The monolithic solution is omitted from the following figures, since it visually overlaps with the analytical solution in all cases, as corroborated by the low values of the NRMS errors listed in the tables.

Table 5: Free evolution NRMS errors with respect to the analytical solution for the two-mass system with and without compensation: undamped and damped system with $T_s = 1$ ms and $T_s = 4$ ms.

	Undamped		**Damped**			
	$T_s = 1$ ms		$T_s = 1$ ms		$T_s = 4$ ms	
	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$
Monolithic	0.027	0.004	0.010	0.010	0.031	0.005
Without compensation	5.159	1.614	0.582	0.171	4.909	1.939
With compensation	0.047	0.038	0.003	0.001	0.032	0.016

The increase of the computational effort due to the higher number of calculations employed during the co-simulation compensation process is evaluated comparing the CPU time resulting from 1000 co-simulations performed without and with the compensation algorithm. A laptop equipped with an Intel i7-950H 2.60 GHz CPU and 16GB RAM has been used. Figure 13 highlights that for the co-simulation without compensation the average CPU time is 1.60 s (dashed line) with $\pm$0.03 s standard deviation (dotted lines) while for the co-simulation with compensation the average CPU time is 1.92 s with $\pm$0.03 s of standard deviation. Hence the CPU time increases about 20% in the studied case.

5.2.2 Free evolution: damped two-mass system with $T_s = 1$ ms

Let us consider the two-mass damped system with $T_s = 1$ ms. In this case, the eigenstructure analysis provided in Section 2.3.2 reveals that the co-simulated system is stable even if no compensation is adopted. However, the spillover on the primary poles deteriorates the accuracy of the co-simulation, leading therefore to unreliable results. The time-domain simulations shown in Fig. 14 and in Fig. 15 confirm the expectations: both f_c and x_2, as well as the mechanical energy, do not diverge, albeit they clearly do not match those sported by the reference model due to spillover on the primary poles.

If the compensator gains provided in Table 4 are applied to correct the co-simulation, sta-

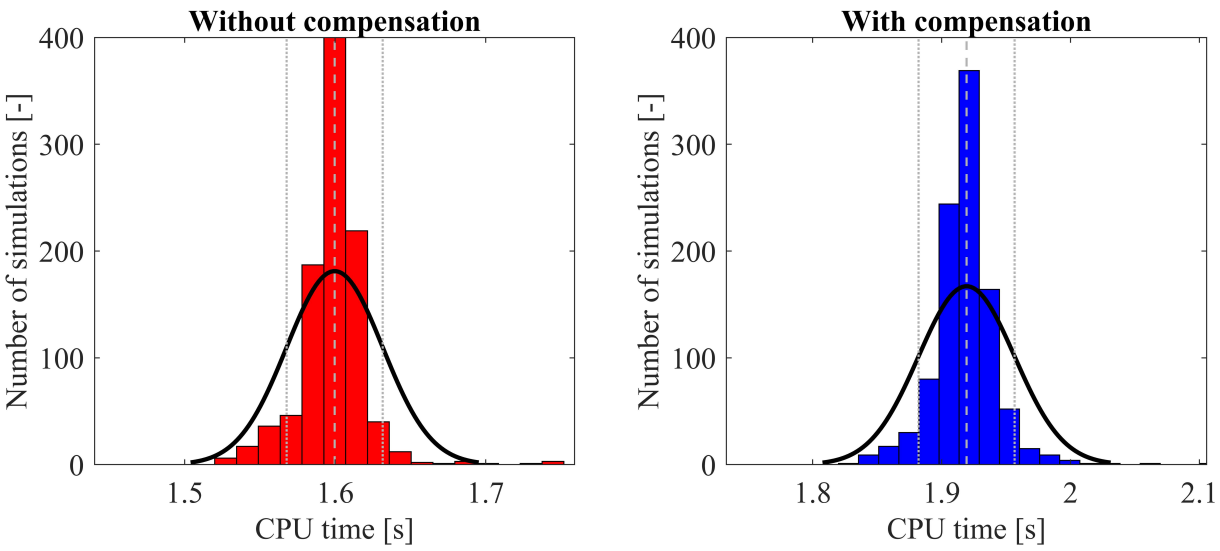


Figure 13: Normal distributions of the co-simulation CPU time for 1000 free-evolution co-simulations with and without compensation algorithm for the two-mass undamped system.

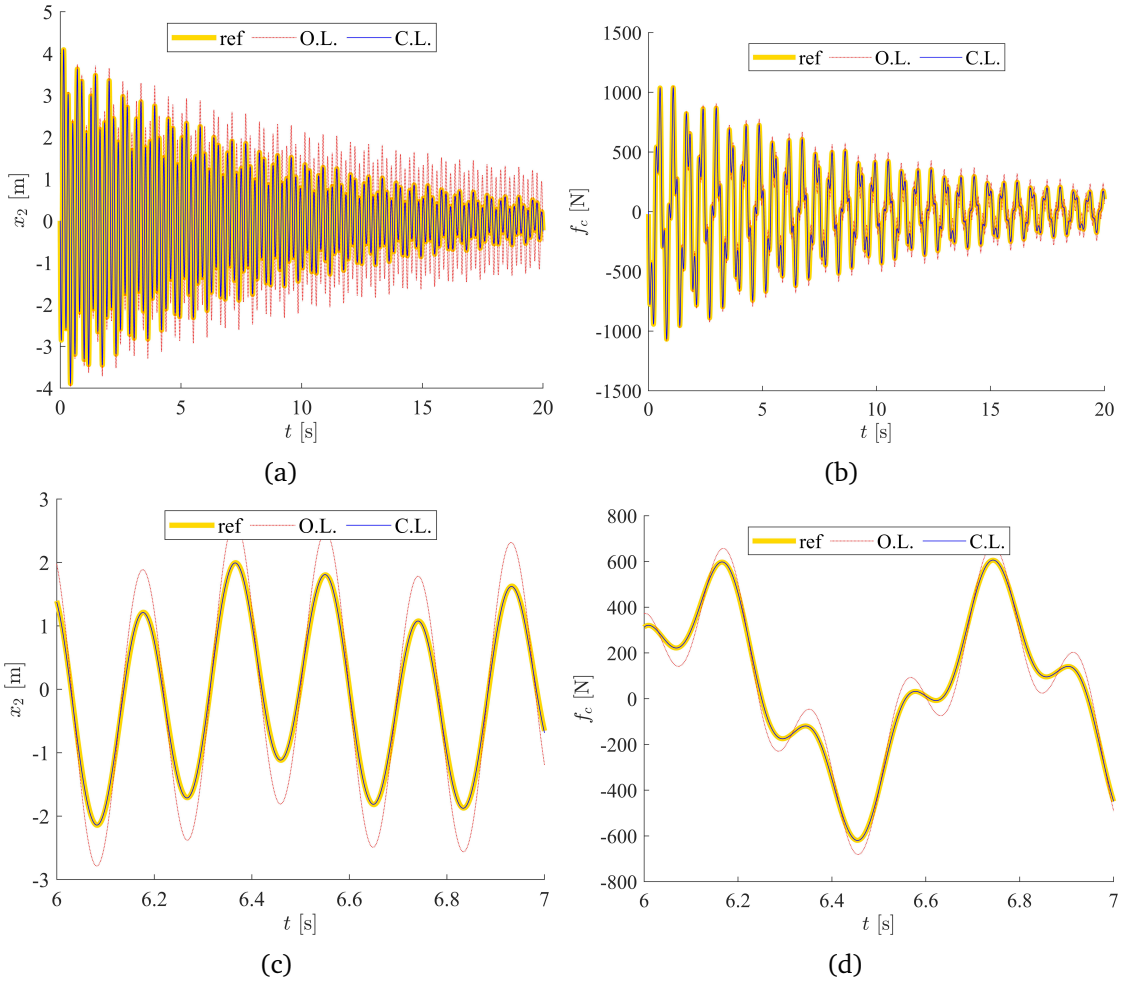
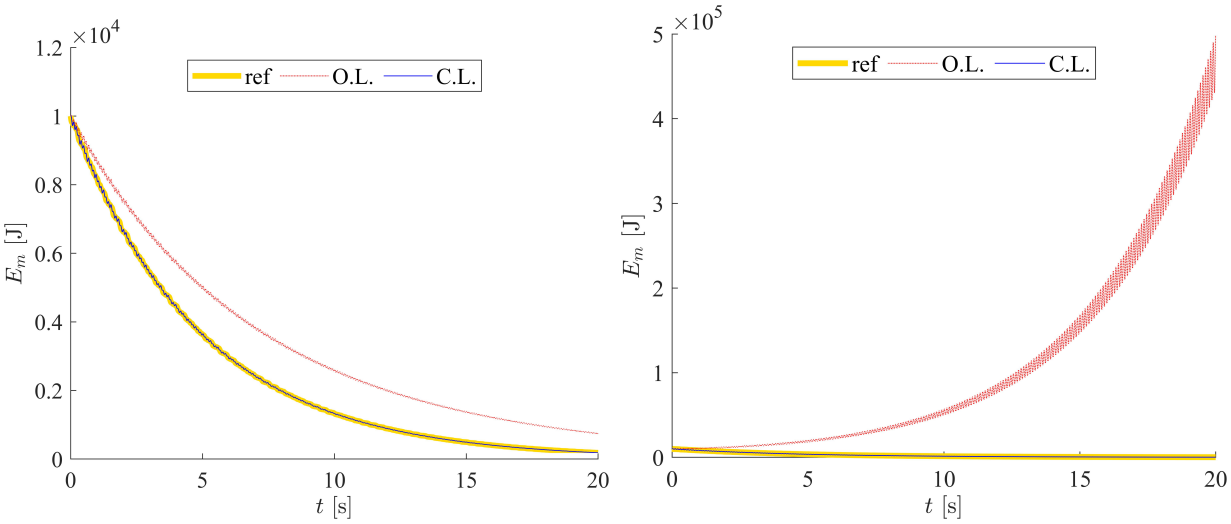


Figure 14: Damped two-mass system with $T_s = 1$ ms time-domain free evolution co-simulation with and without compensation: (a) x_2, (b) f_c, (c) x_2 magnified view, (d) f_c magnified view.



(a) $T_s = 1$ ms (b) $T_s = 4$ ms

Figure 15: Mechanical energy of the damped two-mass system time-domain free evolution co-simulation with and without compensation: (a) $T_s = 1$ ms, (b) $T_s = 4$ ms.

ble and accurate estimates are obtained, as shown in Figs. 14 and 15. As expected, the exact assignment of the eigenstructure makes the corrected co-simulated model almost match those obtained through the reference solution and the monolithic model. This is corroborated by the NRMS errors listed in Table 5: just some small residual errors remain in the compensated co-simulated system, that are hard to see in the time histories in Figs. 14 and 15. These errors are caused by the unavoidable presence of the secondary roots due to the exponential term in $D(s)$, that are not present in the case of the monolithic model.

5.2.3 Free evolution: damped two-mass system with $T_s = 4$ ms

This Section provides the co-simulation results in the challenging scenario of the damped two-mass system simulated using a macro-step size of 4 ms. As shown in Section 2.3.2, the increase of T_s shifts the second pair of primary poles towards the unstable half-plane, thus causing instability. This fact is experienced in the co-simulation of the uncompensated system, that features diverging coupling signals x_2 and f_c (see Fig. 16) and hence diverging mechanical energy (see Fig. 15).

The application of the proposed compensation technique overcomes this issue, leading to a stable behaviour for the co-simulated system that accurately matches the theoretical model, as shown by the time-domain simulations in Figs. 15 and 16, as well as by the low values of e_{NRMS}, as summarized in Table 5.

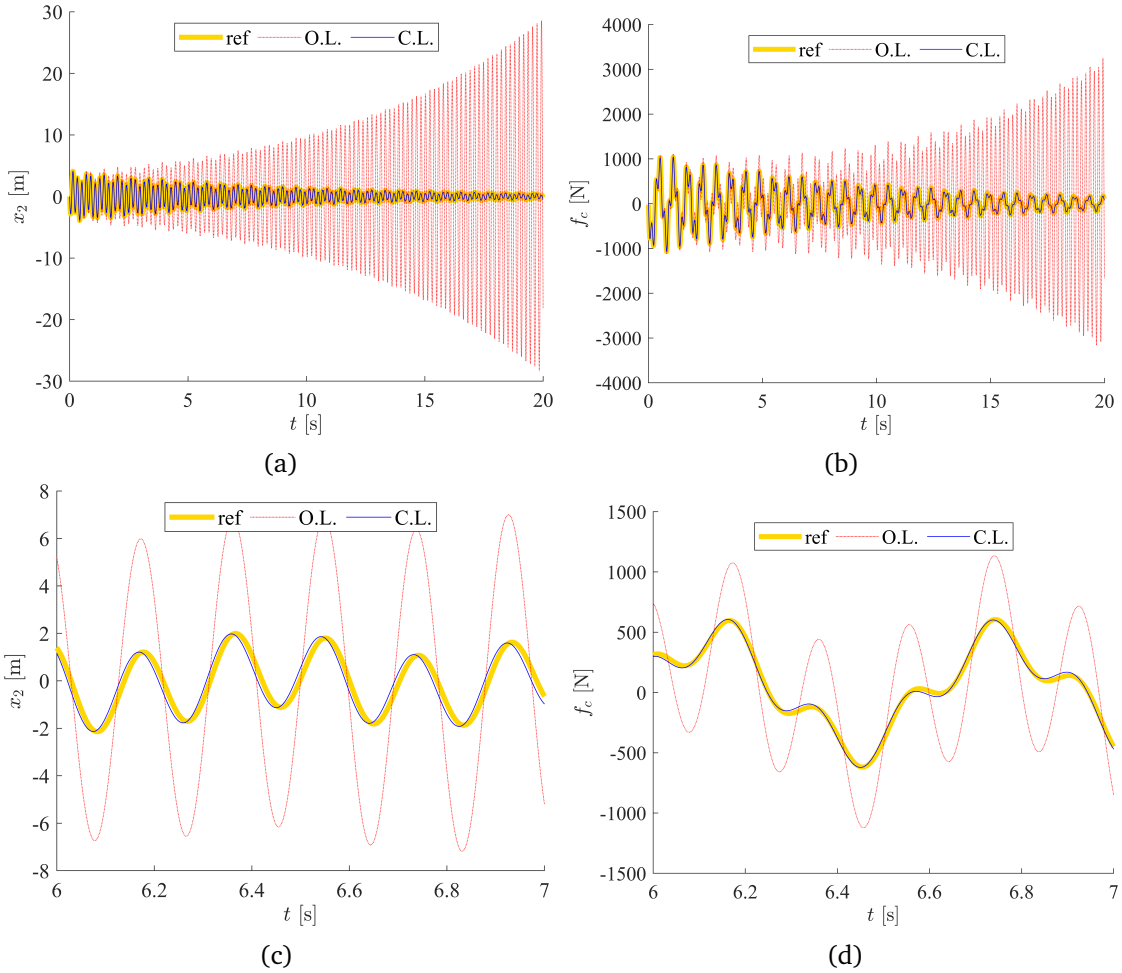


Figure 16: Damped two-mass system with $T_s = 4$ ms time-domain free evolution co-simulation with and without compensation: (a) x_2, (b) f_c, (c) x_2 magnified view, (d) f_c magnified view.

5.3 Co-simulations: forced response

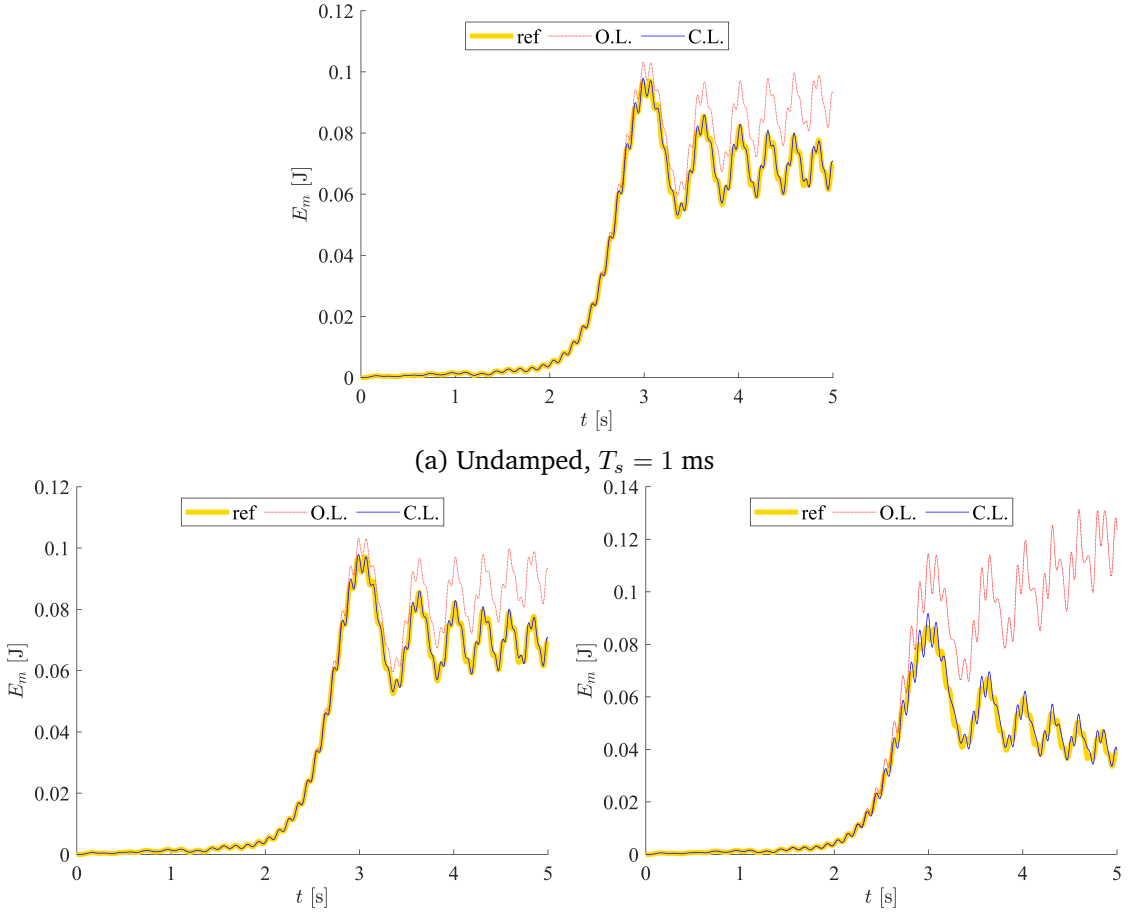
The systems analyzed in the previous Sections were also co-simulated in the presence of a chirp excitation applied to mass m_2, with amplitude 1 N, and frequency ranging from 1 Hz to 10 Hz. The initial conditions for displacement and velocity are set equal to zero. This test is challenging since all the vibrational modes are excited and therefore the frequency response over a broad range can be evaluated.

The reference solution is computed through the numerical integration of the algebraic equation governing the forced response of the system, i.e., [17]:

$$\boldsymbol{z}\left(t\right) = e^{\mathbf{A}t}\boldsymbol{z_0} + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B_{ss}}\boldsymbol{f_e}\left(\tau\right)d\tau \tag{43}$$

where $\mathbf{B_{ss}}$ is the input matrix in the state-space representation and τ denotes the integration

time that spans the interval $[0, t]$.



(a) Undamped, $T_s = 1$ ms

(b) Damped, $T_s = 1$ ms

(c) Damped, $T_s = 4$ ms

Figure 17: Mechanical energy, time-domain forced response co-simulation with and without compensation: (a) undamped system and damped system (b) $T_s = 1$ ms and (c) $T_s = 4$ ms.

The time histories of the mechanical energy are compared in Fig. 17, to provide an abstract representation of all the tests. The instability of the undamped system and of the damped system with $T_s = 4$ ms are properly compensated through the proposed approach, as corroborated by the NRMS errors summarized in Table 6: just small residual errors affect the corrected co-simulated system, because of the unavoidable presence of the secondary poles.

5.4 Co-simulation examples: random parameter systems

The proposed compensation technique is computed automatically, without the need for empirical or trial-and-error tuning: once the analytical model of the system and the transfer function of the co-simulation manager are available, a routine for the off-line computation of the gains

Table 6: Forced response NRMS errors with respect to the reference solution for the two-mass system with and without compensation: undamped and damped system with $T_s = 1$ ms and $T_s = 4$ ms.

	Undamped		**Damped**			
	$T_s = 1$ ms		$T_s = 1$ ms		$T_s = 4$ ms	
	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$	$e^{x_2}_{NRMS}$	$e^{f_c}_{NRMS}$
Monolithic	0.024	0.021	0.040	0.042	0.097	0.095
Without compensation	0.262	0.239	0.155	0.122	0.232	0.182
With compensation	0.150	0.138	0.099	0.084	0.007	0.008

can be executed. The proposed compensation technique can therefore handle arbitrary model parameters. This relevant feature is corroborated in this Section by considering a set of 100 random combinations of the parameters of the mechanical system, generated within the following ranges:

- $m_1, m_2 \in \begin{bmatrix} 1 & 10 \end{bmatrix}$ kg;
- $k_1, k_2, k_c \in \begin{bmatrix} 1 & 1000 \end{bmatrix}$ N/m;
- $\xi_1, \xi_2 \in \begin{bmatrix} 0.001 & 0.1 \end{bmatrix}$;

with $\xi_1 = c_1/2\sqrt{m_1 k_1}$, $\xi_2 = c_2/2\sqrt{m_2 k_2}$ and $c_c \in \begin{bmatrix} 0 & c_1 \end{bmatrix}$. Free evolution from the same initial conditions adopted in Section 5.2 is considered. For all the tests, the macro step-size was set to $T_s = 5$ ms and ZOH extrapolation was used.

The performances of the uncompensated and compensated co-simulated systems are compared in Fig. 18 through the mechanical energy NRMS error in logarithmic scale. It is evident that the compensated co-simulation outperforms the uncompensated, being able to get rid of the perturbation introduced by the co-simulation scheme, the result is highlighted by the average (AVG) NRMS error for 100 simulations displayed in Fig. 18 using a bar plot. The inspection of each co-simulation shows that 13 unstable noncompensated co-simulations arise, as marked through over bar '*' in Fig. 18. In these cases the application of the proposed compensation method enables one to stabilize the co-simulation and to obtain a remarkable accuracy.

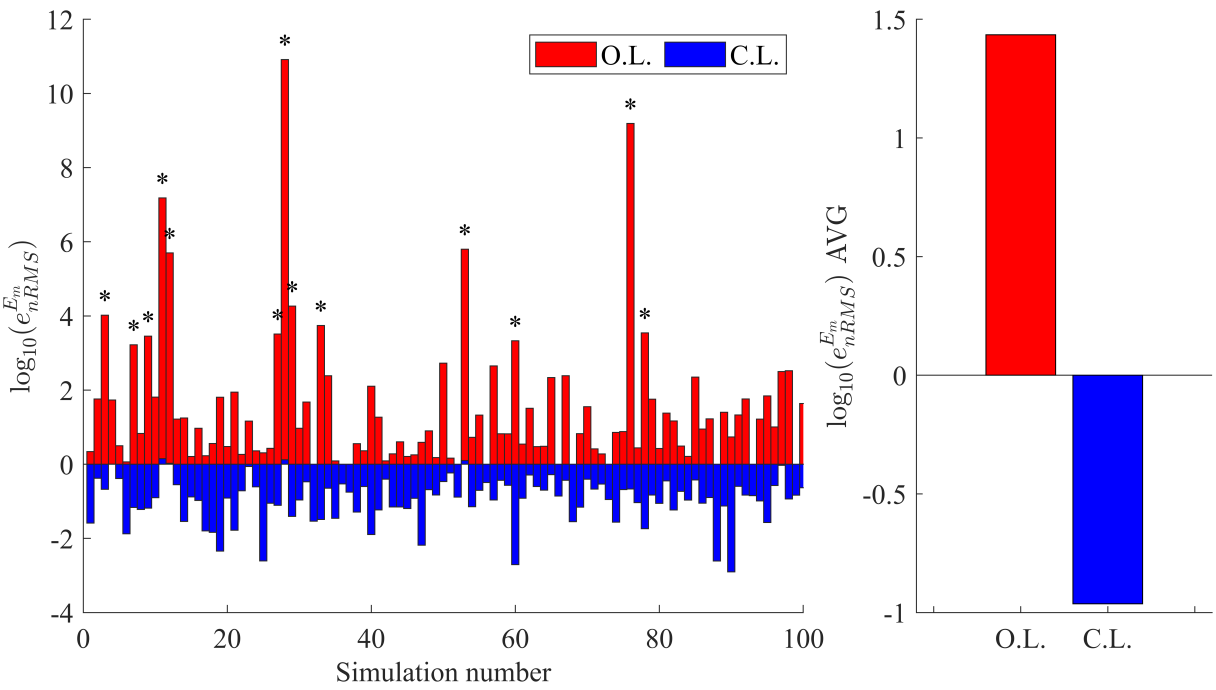


Figure 18: Summary of the mechanical energy NRMS errors in the co-simulation with 100 sets of randomly generated parameter sets. Unstable uncompensated co-simulations are marked by a '*'.

6 Example of application to multi-DOF subsystems

6.1 Eigenstructure assignment

The proposed compensation algorithm is extended in this Section in the case of multi-DOF subsystems.

Let us consider, by means of example, the four-mass system sketched in Fig. 19. Let us assume that both subsystems have two DOFs, exchanging one coupling force f_c and one displacement x_3 through the co-simulation interface.

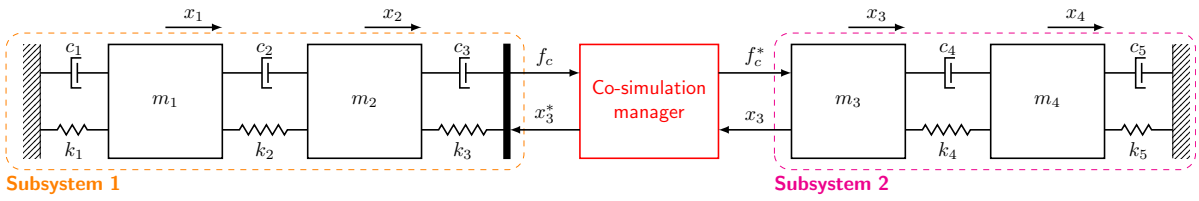


Figure 19: Co-simulated four-mass system with force-displacement coupling.

The theoretical model of the monolithic system is:

$$\mathbf{G_t}(s) = \begin{bmatrix} g_{t,11}(s) & g_{t,12}(s) & 0 & 0 \\ g_{t,21}(s) & g_{t,22}(s) & g_{t,23}(s) & 0 \\ 0 & g_{t,32}(s) & g_{t,33}(s) & g_{t,34}(s) \\ 0 & 0 & g_{t,43}(s) & g_{t,44}(s) \end{bmatrix} \tag{44}$$

where each entry of the dynamic stiffness matrix is:

$$\begin{aligned} g_{t,11}(s) &= s^2 m_1 + s(c_1 + c_2) + k_1 + k_2 \\ g_{t,22}(s) &= s^2 m_2 + s(c_2 + c_3) + k_2 + k_3 \\ g_{t,33}(s) &= s^2 m_3 + s(c_3 + c_4) + k_3 + k_4 \\ g_{t,44}(s) &= s^2 m_4 + s(c_4 + c_5) + k_4 + k_5 \\ g_{t,12}(s) &= g_{t,21} = -(sc_2 + k_2) \\ g_{t,23}(s) &= g_{t,32} = -(sc_3 + k_3) \\ g_{t,34}(s) &= g_{t,43} = -(sc_4 + k_4) \end{aligned} \tag{45}$$

Since $\mathrm{f}_c^*(s) = \mathrm{D}(s)\,\mathrm{f}_c(s)$ and $\mathrm{x}_3^*(s) = \mathrm{D}(s)\,\mathrm{x}_3(s)$, the perturbation introduced by the co-simulation interface is:

$$\mathbf{\Delta G_c}(s) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & (sc_3 + k_3)(1 - \mathrm{D}(s)) & 0 \\ 0 & (sc_3 + k_3)(1 - \mathrm{D}(s)) & (sc_3 + k_3)\left(\mathrm{D}^2(s) - 1\right) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{46}$$

Hence, two independent control forces are here adopted, i.e., $\mathbf{B}$ is selected such that its rank is 2. The two independent compensation forces, denoted u_1 and u_2, are assumed to be applied to mass 2 and mass 3 respectively, and, in accordance with the variables exchanged by the manager, the following definitions are adopted:

$$\begin{cases} \mathrm{u}_1(s) = -\left(\displaystyle\sum_{i=1}^{n_1} \left(s^2 d_i + s f_i + g_i\right) \mathrm{x}_i(s) + \left(s^2 d_{n_1+1} + s f_{n_1+1} + g_{n_1+1}\right) \mathrm{x}_3^*(s) \right) \\ \mathrm{u}_2(s) = -\left(\displaystyle\sum_{\substack{i=1 \\ j=n_1+2}}^{n_1,2n_1+1} \left(s^2 d_j + s f_j + g_j\right) \mathrm{x}_i^*(s) + \sum_{\substack{i=n_1+1 \\ j=2n_1+2}}^{n_2,2n_1+n_2+1} \left(s^2 d_j + s f_j + g_j\right) \mathrm{x}_i(s) \right) \end{cases} \tag{47}$$

It is worth mentioning that the proposed control scheme is not a full state feedback, since u_1 does not exploit any information on x_4. Indeed, on the one hand, x_4 is not exchanged through the manager; on the other hand, subsystem 2 does not send back any reaction force to subsystem 1 through the manager, and hence the "trick" adopted in the two-mass system to perform full state feedback is not allowed. In contrast, the "output feedback" compensator in the form of Eq. (47) can be implemented with the same approach proposed in Section 5.1, leading to the following correction (with $\mathrm{r_j}\left(s\right) = s^2 d_j + s f_j + g_j$):

$$\mathbf{B\Delta G_R}\left(s\right) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \mathrm{r}_1\left(s\right) & \mathrm{r}_2\left(s\right) & \mathrm{r}_3\left(s\right)\mathrm{D}\left(s\right) & 0 \\ \mathrm{r}_4\left(s\right)\mathrm{D}\left(s\right) & \mathrm{r}_5\left(s\right)\mathrm{D}\left(s\right) & \mathrm{r}_6\left(s\right) & \mathrm{r}_7\left(s\right) \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{48}$$

By formulating the eigenproblem of the compensated system, as provided in Eq. (21), the linear system of Eq. (27) is formulated to compute the gains:

$$\begin{aligned}
\mathbf{L_{a,i}} &= \begin{bmatrix} \lambda_i^2\mathbf{w}_i^{(1)} & \lambda_i^2\mathbf{w}_i^{(2)} & \lambda_i^2\mathbf{w}_i^{(3)}\mathrm{D}\left(\lambda_i\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_i^2\mathbf{w}_i^{(1)}\mathrm{D}\left(\lambda_i\right) & \lambda_i^2\mathbf{w}_i^{(2)}\mathrm{D}\left(\lambda_i\right) & \lambda_i^2\mathbf{w}_i^{(3)} & \lambda_i^2\mathbf{w}_i^{(4)} \end{bmatrix} \\
\mathbf{L_{v,i}} &= \begin{bmatrix} \lambda_i\mathbf{w}_i^{(1)} & \lambda_i\mathbf{w}_i^{(2)} & \lambda_i\mathbf{w}_i^{(3)}\mathrm{D}\left(\lambda_i\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_i\mathbf{w}_i^{(1)}\mathrm{D}\left(\lambda_i\right) & \lambda_i\mathbf{w}_i^{(2)}\mathrm{D}\left(\lambda_i\right) & \lambda_i\mathbf{w}_i^{(3)} & \lambda_i\mathbf{w}_i^{(4)} \end{bmatrix} \\
\mathbf{L_{d,i}} &= \begin{bmatrix} \mathbf{w}_i^{(1)} & \mathbf{w}_i^{(2)} & \mathbf{w}_i^{(3)}\mathrm{D}\left(\lambda_i\right) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{w}_i^{(1)}\mathrm{D}\left(\lambda_i\right) & \mathbf{w}_i^{(2)}\mathrm{D}\left(\lambda_i\right) & \mathbf{w}_i^{(3)} & \mathbf{w}_i^{(4)} \end{bmatrix} \\
\begin{Bmatrix} n_{1,i} \\ n_{2,i} \end{Bmatrix} &= -\begin{Bmatrix} \left(\lambda_i c_3 + k_3\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right)\mathbf{w}_i^{(2)} \\ \left(\lambda_i c_3 + k_3\right)\left(1 - \mathrm{D}\left(\lambda_i\right)\right)\mathbf{w}_i^{(2)} + \left(\lambda_i c_3 + k_3\right)\left(D^2\left(\lambda_i\right) - 1\right)\mathbf{w}_i^{(3)} \end{Bmatrix} \\
\mathbf{d} &= \begin{Bmatrix} d_1 & \dots & d_7 \end{Bmatrix}^T, \quad \mathbf{f} = \begin{Bmatrix} f_1 & \dots & f_7 \end{Bmatrix}^T, \quad \mathbf{g} = \begin{Bmatrix} g_1 & \dots & g_7 \end{Bmatrix}^T
\end{aligned} \tag{49}$$

The rank analysis of the linear system reveals that this control architecture satisfies the assignability condition.

6.2 Co-simulation examples

Let us consider the following parameters for the simply-connected chain of the four-mass system sketched in Fig. 19: $m_1 = \frac{1}{2}m_2 = \frac{1}{3}m_3 = \frac{1}{4}m_4 = 1$ kg, $k_1 = \frac{1}{2}k_2 = \frac{1}{3}k_3 = \frac{1}{4}k_4 = \frac{1}{5}k_5 = 100$ N/m, $c_1 = \frac{1}{2}c_2 = \frac{1}{3}c_3 = \frac{1}{4}c_4 = \frac{1}{5}c_5 = 0.01$ Ns/m. The macro-step size T_s is 3 ms.

The eigenvalue analysis of the co-simulated system, summarized in Table 7, highlights that the uncompensated co-simulated system will be unstable due to the presence of three pole pairs that lie in the right half of the complex plane, as shown in Fig. 20. The instability of the co-simulated system without compensation is confirmed also by the negative damping ratios obtained for the higher frequency pole pairs as summarized in Table 8.

Table 7: Primary poles of the four-mass system: monolithic and co-simulated without and with compensation.

Eigenvalue	**Monolithic**	**Co-simulated**	**Co-simulated compensated**
$\lambda_{1,2}$	$-0.002 \pm 6.586j$	$-0.011 \pm 6.586j$	$-0.002 \pm 6.586j$
$\lambda_{3,4}$	$-0.009 \pm 13.120j$	$0.031 \pm 13.120j$	$-0.009 \pm 13.120j$
$\lambda_{5,6}$	$-0.017 \pm 18.206j$	$0.017 \pm 18.206j$	$-0.017 \pm 18.206j$
$\lambda_{7,8}$	$-0.023 \pm 21.479j$	$0.063 \pm 21.479j$	$-0.023 \pm 21.479j$

Table 8: Modal parameters of the four mass system: monolithic and co-simulated without and with compensation.

	$\omega_{n_{1,2}}$[rad/s]	$\xi_{1,2}$	$\omega_{n_{3,4}}$[rad/s]	$\xi_{3,4}$	$\omega_{n_{5,6}}$[rad/s]	$\xi_{5,6}$	$\omega_{n_{7,8}}$[rad/s]	$\xi_{7,8}$
Monolithic	6.59	0.0003	13.1	0.0007	18.2	0.0009	21.5	0.0011
Co-simulated	6.59	0.0017	13.1	-0.0024	18.2	-0.0009	21.5	-0.0029
Co-simulated compensated	6.59	0.0003	13.1	0.0007	18.2	0.0009	21.5	0.0011

The application of the eigenstructure assignment strategy compensates for the perturbation caused by the co-simulation interface, indeed the natural frequencies and damping ratios of the co-simulated system with compensation match those of the monolithic system as summarized in Table 8. Figure 20 shows that the primary poles in the compensated co-simulation are shifted back to their theoretical locations in the complex plane, i.e., those of the monolithic system.

The effectiveness of the proposed compensation strategy in the presence of multi-DOF subsystems is corroborated by the co-simulation of the free evolution of the system from a sample set of initial conditions, that includes null initial displacement and $\dot{\boldsymbol{x}}_{\mathbf{0}} = \begin{Bmatrix} 1 & -2 & 3 & -4 \end{Bmatrix}^T$

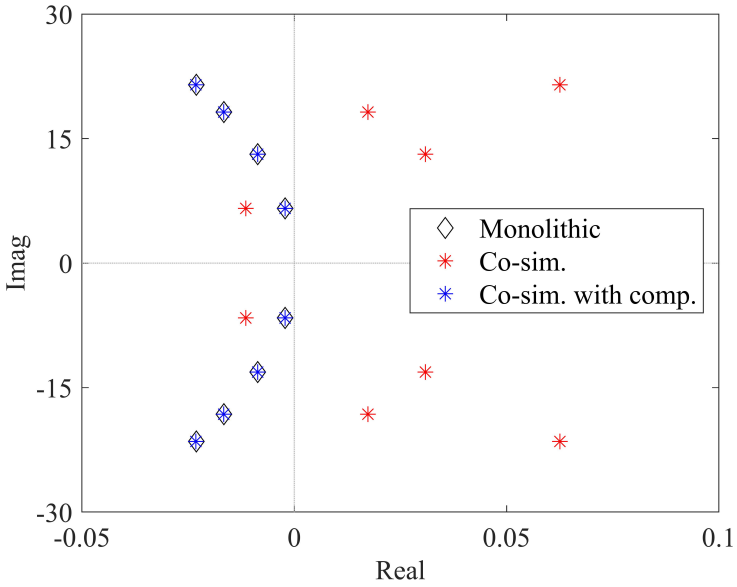


Figure 20: Primary poles of the monolithic and co-simulated four-mass system without and with compensation.

m/s.

The mechanical energy of the system is shown in Fig. 21 and confirms that the original co-simulated system is unstable. Once the compensation is implemented, the co-simulated system is stabilized and its dynamics matches that of its theoretical counterpart as corroborated by the coupling signals x_3 and f_c shown in Fig. 22 and by the NRMS errors listed in Table 9.

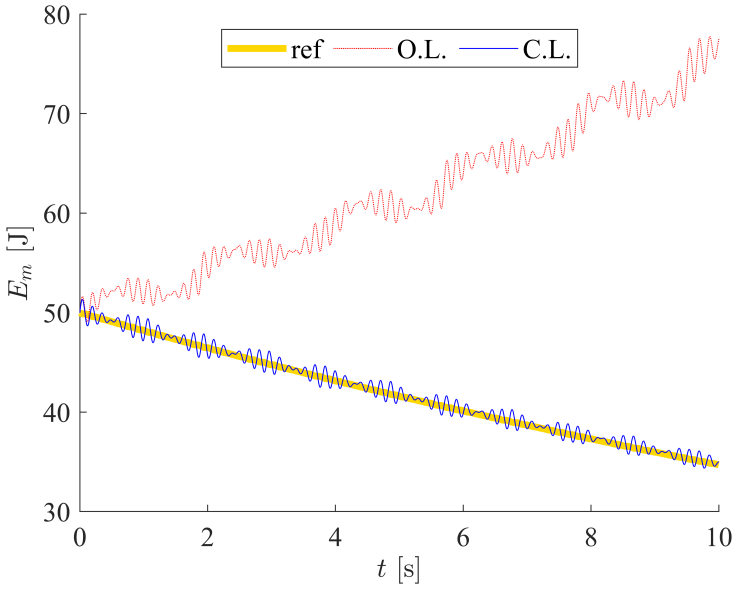


Figure 21: Mechanical energy in the co-simulation of the four-mass system with and without compensation.

The computational effort of the co-simulation with compensation is evaluated comparing

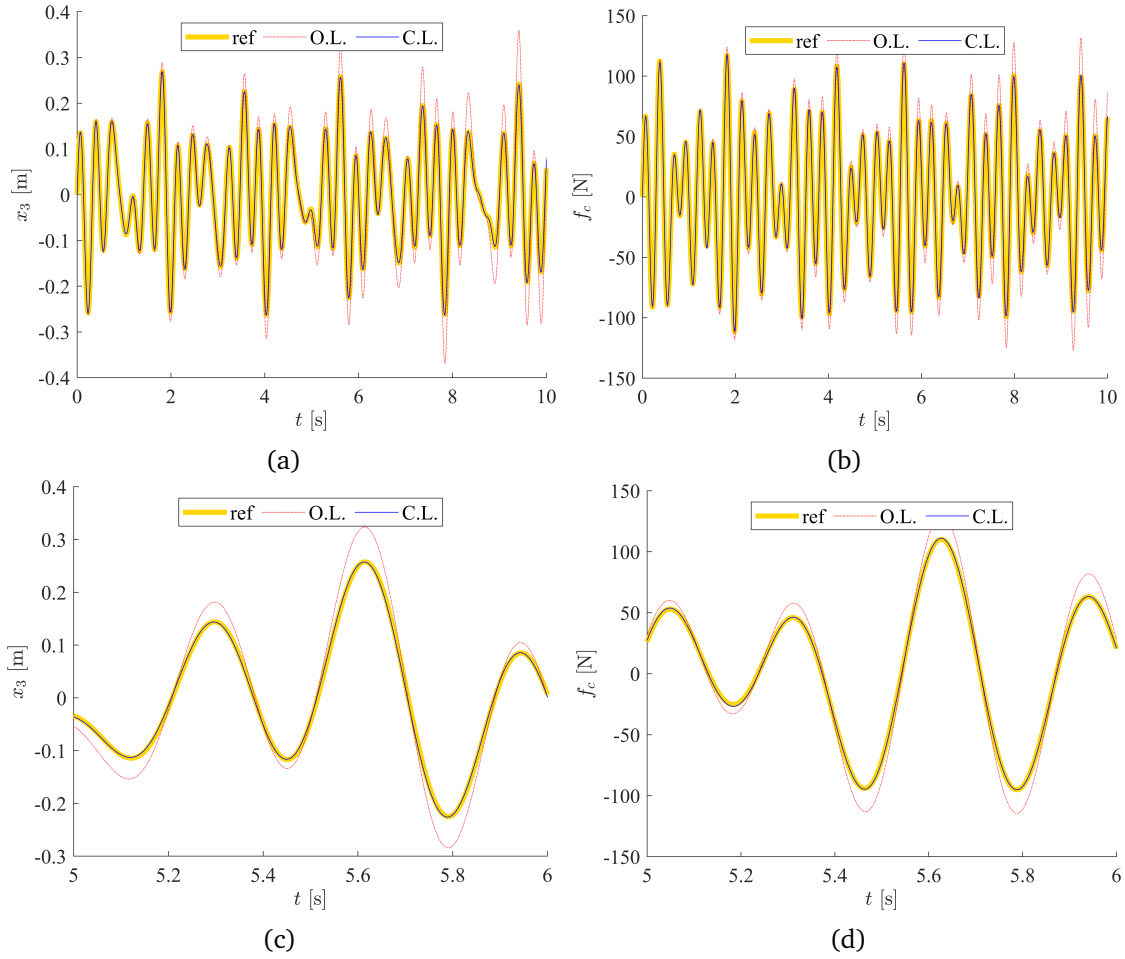


Figure 22: Four-mass system co-simulation with and without compensation: (a) x_3, (b) f_c, (c) x_3 magnified view, (d) f_c magnified view.

Table 9: NRMS errors with respect to the analytical solution for the four-mass system with and without compensation.

	$e^{x_3}_{NRMS}$	$e^{f_c}_{NRMS}$
Monolithic	0.024	0.053
Co-simulated	1.906	0.230
Co-simulated compensated	0.136	0.046

the CPU time resulting from 1000 co-simulations performed without and with the compensation algorithm for this test-case as done in Section 5.2 for the two-mass system. The CPU times shown in Figure 13 evidence that for the co-simulation without compensation the average CPU time is 0.35 s with $\pm$0.03 s standard deviation while for the co-simulation with compensation the average CPU time is 0.41 s with $\pm$0.04 s of standard deviation. Hence the CPU time increases about 17%. Clearly, the CPU time is smaller with respect to the two mass system since the system is simulated for 10 s and with a larger step size.

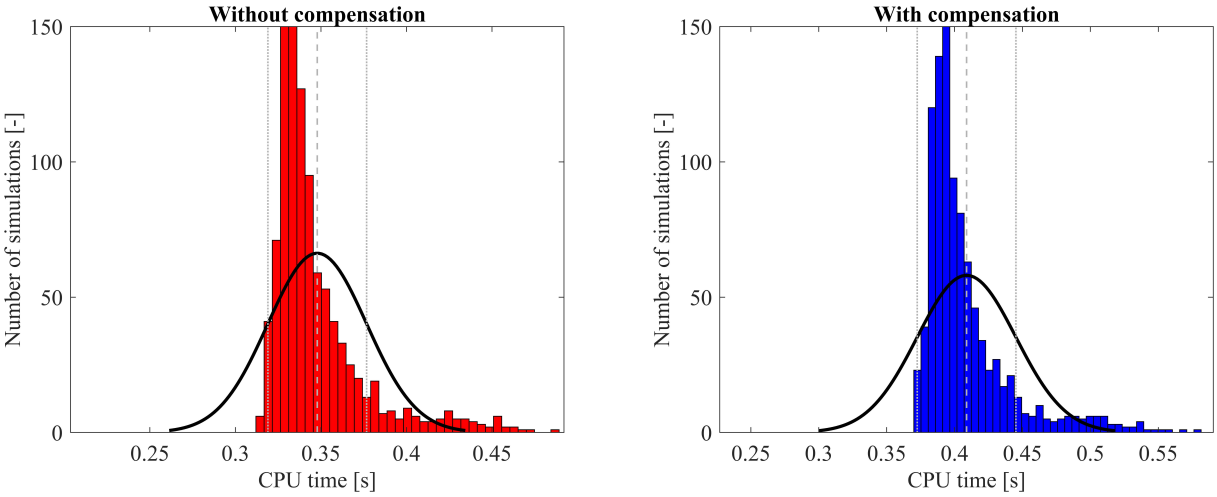


Figure 23: Normal distributions of the co-simulation CPU time for 1000 free-evolution co-simulations with and without compensation algorithm for the four-mass system.

7 Conclusions

In this paper the stability and accuracy of explicit co-simulation are discussed. The dynamics of co-simulated systems in explicit Jacobi schemes is formulated taking into account the effect of the signal exchange between subsystems through the co-simulation manager. This yields a model of the co-simulated system that explicitly expresses the perturbation with respect to its

monolithic counterpart introduced by the signal coupling interface. The adopted mathematical formulation enables one to exploit a novel approach based on the analysis of the co-simulated system eigenstructure, which in turn allows the evaluation of the alteration of the system dynamics due to the spillover of the system eigenvalues and eigenvectors. In addition, since co-simulation is expressed as the closed-loop interconnection between subsystems, the presence of time-delays due to the signal exchange process introduces an infinite number of secondary poles, the latent roots, which may compromise co-simulation stability. The effectiveness of the analysis of the system eigenstructure to predict instability and inaccuracy during co-simulation is corroborated by some numerical examples performed on a widely adopted benchmark taken from the literature.

A novel co-simulation compensation strategy, based on the linear control theory, is adopted to perform the eigenstructure assignment, i.e., to compensate for the perturbation on the system poles (i.e., natural frequencies and damping) and eigenvector (i.e., mode shapes) due to the co-simulation. The compensator is designed by considering an ad-hoc developed condition to verify the assignability of the system eigenstructure through the designed compensation architecture.

The proposed compensation strategy effectiveness is assessed by several numerical co- simulations performed on a two-mass system with different parameters and by varying both the system initial conditions as well as the excitation force. Several macro-step sizes were used in the verification process as well. The method has been further extended to compensate the co-simulation of linear mechanical systems with several degrees of freedom and its effectiveness in this scenario is assessed by a numerical example that consists of a four-mass system.

The generalization of the proposed approach for its use in a wider scope of problems represents a currently open line of research. The method could be extended to nonlinear systems by means of piecewise linearization of the subsystem dynamics. It would also be possible to use the proposed compensation solution when the internals of the subsystems are not disclosed, using identification techniques and preliminary test runs of the subsystems. These improvements will make it possible to expand the applicability of the method to wider areas of multi-domain co-simulation.

Acknowledgments

F. González acknowledges the support of the Ministry of Economy of Spain through the Ramón y Cajal research program, contract no. RYC-2016-20222, and of the Government of Galicia through grant ED431F2021/04.

References

[1] Adamson, L., Fichera, S., Mottershead, J.: Receptance-based robust eigenstructure assignment. Mechanical Systems and Signal Processing **140**, 106,697 (2020). DOI 10.1016/j.ymssp.2020.106697

[2] Andry, A.N., Shapiro, E.Y., Chung, J.: Eigenstructure assignment for linear systems. IEEE Transactions on Aerospace and Electronic Systems **AES-19**(5), 711–729 (1983). DOI 10.1109/TAES.1983.309373

[3] Apkarian, P., Tuan, H.D., Bernussou, J.: Continuous-time analysis, eigenstructure assignment, and H_2 synthesis with enhanced linear matrix inequalities (LMI) characterizations. IEEE Transactions on Automatic Control **46**(12), 1941–1946 (2001). DOI 10.1109/9.975496

[4] Araújo, J.M., Bettega, J., Dantas, N.J., Dórea, C.E., Richiedei, D., Tamellin, I.: Vibration control of a two-link flexible robot arm with time delay through the robust receptance method. Applied Sciences **11**(21), 9907 (2021). DOI 10.3390/app11219907

[5] Araújo, J.M., Dórea, C.E., Gonçalves, L.M., Datta, B.N.: State derivative feedback in second-order linear systems: A comparative analysis of perturbed eigenvalues under coefficient variation. Mechanical Systems and Signal Processing **76**, 33–46 (2016). DOI 10.1016/j.ymssp.2016.02.014

[6] Arnold, M., Clauss, C., Schierz, T.: Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation v2.0. Archive of Mechanical Engineering **60**(1), 75–94 (2013). DOI 10.2478/meceng-2013-0005

[7] Belotti, R., Richiedei, D.: Pole assignment in vibrating systems with time delay: An approach embedding an a-priori stability condition based on linear matrix inequality. Me-

chanical Systems and Signal Processing **137**, 106,396 (2020). DOI 10.1016/j.ymssp.2019.106396

[8] 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), 185–200 (2017). DOI 10.1177/0037549716684026

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

[10] Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: Macro-step-size selection and monitoring of the coupling error for weak coupled subsystems in the frequency-domain. In: S. Idelsohn, M. Papadrakis, B. Schrefler (eds.) International Conference on Computational Methods for Coupled Problems in Science and Engineering, pp. 1009–1020. Ibiza, Spain (2013)

[11] Benedikt, M., Watzenig, D., Zehetner, J., Hofer, A.: NEPCE – a nearly energy-preserving coupling element for weak-coupled problems and co-simulations. In: S. Idelsohn, M. Papadrakis, B. Schrefler (eds.) International Conference on Computational Methods for Coupled Problems in Science and Engineering, pp. 1021–1032. Ibiza, Spain (2013)

[12] Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Clauss, C., Elmqvist, H., Junghanns, A., Mauss, J., Monteiro, M., Neidhold, T., Neumerkel, D., Olsson, H., Peetz, J.V., Wolf, S.: The Functional Mockup Interface for tool independent exchange of simulation models. In: Proceedings of the 8th International Modelica Conference; Dresden; Germany, pp. 105–114 (2011). DOI 10.3384/ecp11063105

[13] Busch, M.: Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **96**(9), 1061–1081 (2016). DOI 10.1002/zamm.201500196

[14] Chen, W., Ran, S., Wu, C., Jacobson, B.: Explicit parallel co-simulation approach: analysis and improved coupling method based on h-infinity synthesis. Multibody System Dynamics **52**, 255–279 (2021). DOI 10.1007/s11044-021-09785-x

[15] Chu, E., Datta, B.: Numerically robust pole assignment for second-order systems. International Journal of Control **64**(6), 1113–1127 (1996). DOI 10.1080/00207179608921677

[16] Cleveland, W.: First-order-hold interpolation digital-to-analog converter with application to aircraft simulation. NASA TN D-8331, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, Washington, D. C. (1976)

[17] Datta, B.: Numerical methods for linear control systems, vol. 1. Academic Press (2004)

[18] Farhati, S., Naoui, S.B.H.A., Abdelkrim, M.N.: H_{inf} loop shaping control of time delay systems. In: 2011 International Conference on Communications, Computing and Control Applications (CCCA), pp. 1–6. IEEE (2011). DOI 10.1109/CCCA.2011.6031512

[19] Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: a survey. ACM Computing Surveys **51**(3), 1–33 (2018). DOI 10.1145/3179993

[20] González, F., Arbatani, S., Mohtat, A., Kövecses, J.: Energy-leak monitoring and correction to enhance stability in the co-simulation of mechanical systems. Mechanism and Machine Theory **131**, 172–188 (2019). DOI 10.1016/j.mechmachtheory.2018.09.007

[21] 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 System Dynamics **25**(4), 461–483 (2011). DOI 10.1007/s11044-010-9234-7

[22] Haid, T., Stettinger, G., Watzenig, D., Benedikt, M.: A model-based corrector approach for explicit co-simulation using subspace identification. In: The 5th Joint International Conference on Multibody System Dynamics. Lisbon, Portugal (2018)

[23] Haid, T., Watzenig, D., Stettinger, G.: Analysis of the model-based corrector approach for explicit cosimulation. Multibody System Dynamics **Online first** (2022). DOI 10.1007/s11044-022-09822-3

[24] Kraft, J., Klimmek, S., Meyer, T., Schweizer, B.: Implicit co-simulation and solver-coupling: Efficient calculation of interface-Jacobian and coupling sensitivities/gradients. Journal of Computational and Nonlinear Dynamics **17**(4), CND–21–1063 (2022). DOI 10.1115/1.4051823

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

[26] Li, P., Yuan, Q.: Influence of coupling approximation on the numerical stability of explicit co-simulation. Journal of Mechanical Science and Technology **34**, 2289–2298 (2020). DOI 10.1007/s12206-020-0504-x

[27] Meyer, T., Li, P., Lu, D., Schweizer, B.: Implicit co-simulation method for constraint coupling with improved stability behavior. Multibody System Dynamics **44**(2), 135–161 (2018). DOI 10.1007/s11044-018-9632-9

[28] 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. Archive of Mechanical Engineering **58**(2), 223–243 (2011). DOI 10.2478/v10180-011-0016-4

[29] Oberschelp, O., Vöcking, H.: Multirate simulation of mechatronic systems. In: Proceedings of the IEEE International Conference on Mechatronics, pp. 404–409. Istanbul, Turkey (2004). DOI 10.1109/icmech.2004.1364473

[30] Olgac, N., Sipahi, R.: An exact method for the stability analysis of time-delayed linear time-invariant (lti) systems. IEEE Transactions on Automatic Control **47**(5), 793–797 (2002). DOI 10.1109/TAC.2002.1000275

[31] Ouyang, H., Richiedei, D., Trevisani, A.: Pole assignment for control of flexible link mechanisms. Journal of Sound and Vibration **332**(12), 2884–2899 (2013). DOI 10.1016/j.jsv.2013.01.004

[32] Peiret, A., González, F., Kövecses, J., Teichmann, M., Enzenhoefer, A.: Model-based coupling for co-simulation of robotic contact tasks. IEEE Robotics and Automation Letters **5**(4), 5756–5763 (2020). DOI 10.1109/LRA.2020.3010204

[33] Rahikainen, J., González, F., Naya, M.Á.: An automated methodology to select functional co-simulation configurations. Multibody System Dynamics **48**(1), 79–103 (2020). DOI 10.1007/s11044-019-09696-y

[34] Ram, Y.M., Mottershead, J.E.: Receptance method in active vibration control. AIAA journal **45**(3), 562–567 (2007). DOI 10.2514/1.24349

[35] Richiedei, D., Tamellin, I.: Active control of linear vibrating systems for antiresonance assignment with regional pole placement. Journal of Sound and Vibration **494**, 115,858 (2021). DOI 10.1016/j.jsv.2020.115858

[36] Richiedei, D., Tamellin, I., Trevisani, A.: Unit-rank output feedback control for antiresonance assignment in lightweight systems. Mechanical Systems and Signal Processing **164**, 108,250 (2022). DOI 10.1016/j.ymssp.2021.108250

[37] Richiedei, D., Trevisani, A.: Simultaneous active and passive control for eigenstructure assignment in lightly damped systems. Mechanical Systems and Signal Processing **85**, 556–566 (2017). DOI 10.1016/j.ymssp.2016.08.046

[38] Rodríguez, B., Rodríguez, A.J., Sputh, B., Pastorino, R., Naya, M.Á., González, F.: Energy-based monitoring and correction to enhance the accuracy and stability of explicit co-simulation. Multibody System Dynamics **Online first** (2022). DOI 10.1007/s11044-022-09812-5

[39] 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. Engineering with Computers **33**(3), 607–620 (2017). DOI 10.1007/s00366-016-0492-8

[40] Samin, J.C., Brüls, O., Collard, J.F., Sass, L., Fisette, P.: Multiphysics modeling and optimization of mechatronic multibody systems. Multibody System Dynamics **18**(3), 345–373 (2007). DOI 10.1007/s11044-007-9076-0

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

[42] Schweizer, B., Lu, D.: Predictor/corrector co-simulation approaches for solver coupling with algebraic constraints. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **95**(9), 911–938 (2015). DOI 10.1002/zamm.201300191

[43] Stettinger, G., Horn, M., Benedikt, M., Zehetner, J.: Model-based coupling approach for non-iterative real-time co-simulation. In: 2014 European Control Conference (ECC), pp. 2084–2089. Strasbourg, France (2014). DOI 10.1109/ECC.2014.6862242

[44] Tsai, S.H., Ouyang, H., Chang, J.Y.: Inverse structural modifications of a geared rotor-bearing system for frequency assignment using measured receptances. Mechanical Systems and Signal Processing **110**, 59–72 (2018). DOI 10.1016/j.ymssp.2018.03.008

[45] Van Loan, C.F., Golub, G.: Matrix computations, 3 edn. The Johns Hopkins University Press, Baltimore and London (1996)

[46] Zhang, J., Ouyang, H., Yang, J.: Partial eigenstructure assignment for undamped vibration systems using acceleration and displacement feedback. Journal of Sound and Vibration **333**(1), 1–12 (2014). DOI 10.1016/j.jsv.2013.08.040