ZAMM · Z. Angew. Math. Mech. 1–20 (2017) / **DOI** 10.1002/zamm.201500129

A new approach for force-displacement co-simulation using kinematic coupling constraints

Fabio Schneider[1,*], **Michael Burger**[1], **Martin Arnold**[2], **and Bernd Simeon**[3]

[1] Fraunhofer Institute for Industrial Mathematics, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
[2] Martin Luther University Halle-Wittenberg, Institute of Mathematics, 06099 Halle (Saale), Germany
[3] TU Kaiserslautern, Department of Mathematics, Paul-Ehrlich-Straße, 67663 Kaiserslautern, Germany

Received 8 May 2015, revised 4 June 2016, accepted 17 February 2017
Published online 10 April 2017

Key words Kinematic coupling, algebraic constraint, co-simulation.

The separate numerical simulation of interacting subsystems, i.e. the co-simulation of subsystems, is widespread in advanced system simulation. Classically, mechanical subsystems are coupled by artificial stiffnesses, which often are unknown. In this paper, we present a coupling approach that uses no artificial stiffness, but a kinematic coupling constraint for the co-simulation of different subsystems. The algebraic constraint leads to corresponding constraint forces that only act in one direction of the coupling to avoid drift-off problems, while displacements are given in the opposite direction. Moreover, a strategy for an efficient computation of approximated constraint forces is provided. For the stability of the approach, we consider zero-stability as well as stability for finite time steps. Convergence is analyzed with a 2-mass spring-damper model, where the analytical solution is known. As application, we use our approach to simulate the more complex problem of a cable model, that is embedded in a multibody system.

1 Introduction

In applied numerical system simulation, it becomes more and more important to simulate several subsystems from different physical domains, available in different software tools, equipped with different numerical solvers, at the same time to handle their dynamic interaction. For instance, in mechanical or mechatronical applications, especially in vehicle engineering, this is an important aspect. The overall system is typically modelled as rigid multibody system, and one has to take into account several subsystems, e.g., flexible components, hydraulic actuators, rubber bearings or control devices. All these elements influence significantly the system's behaviour.

Another wide field of application are fluid-structure interactions, e.g. in aerospace engineering, as discussed in [11]. Depending on the physical effects of a fluid one incorporates or neglects in the simulation, the resulting equations and possible discretization techniques strongly differ making this topic very special.

From the numerical point of view, it is especially challenging that the subsystems usually differ in their time constants and scales, in their formulation as well as in their complexity and, thus, in their computation time. Therefore, it would be difficult to solve the entire system with one monolithic numerical solver in an efficient way. Especially, when the subsystems are available in different commercial software tools, this might be impossible. In contrast, modern co-simulation strategies aim at using established problem- and domain-specific numerical solvers for each subsystem and exchange only a small number of interface quantities between the subsystems at certain time points, cf. [4–7, 13, 17, 18]. In order to support the co-simulation of dynamic models, an interface standard FMI (Functional Mock-up Interface) was developed [1].

The main application, this paper is focused on, are flexible multibody systems. For these systems, certain components have to be modelled flexible. A detailed discussion can be found e.g. in [19]. Classically, they are formulated with linear finite element models and a modal reduction approach, see e.g. [8], is used to add the reduced degrees of freedom of the flexible structure to the multibody system. The flexible states, i.e. the deformations, are expressed in a floating frame of reference that moves with the coupled rigid body of the multibody system. The resulting monolithic system, which incorporates flexible bodies, is solved by the multibody system integrator. This approach, however, becomes inaccurate for large deformations of the flexible structure, which occur, e.g., in the assembly simulation of cables and hoses.

To overcome the drawback, one can follow a co-simulation approach as mentioned before. That is, the flexible component model is coupled via a co-simulation interface to the multibody system, both systems can be solved with their own well

* Corresponding author, e-mail: fabio.schneider@itwm.fraunhofer.de, Phone: +49 631 31600 4730

suited integrator and they only have to exchange interface quantities at certain time points. Additionally, this is applicable to arbitrary flexible component models, without any restrictions.

In this contribution, we consider a kinematic coupling, i.e., a coupling of two mechanical subsystems, formulated as ordinary differential equations, via an arbitrary kinematic joint. This kind of coupling is especially suited to combine rigid multibody systems and flexible components. We propose a strategy to efficiently derive joint reaction forces which are used as interface quantities. To do so, we formally start with a monolithic description of the problem and formulate an algebraic constraint for the kinematic joint. From that, the joint reaction constraint forces are derived. An exact computation of the constraint forces by solving the large monolithic index-1-system is computationally very expensive. Therefore, we introduce a strategy to approximately compute the constraint forces in a very efficient way. Potential instabilities resulting from this will be analyzed. The co-simulation proceeds in a force-displacement scheme, where the derived constraint forces and prescribed displacements, which are computed via the kinematic constraint, are exchanged. This force-displacement co-simulation procedure strongly differs from the classical one, see e.g. [5]. A different coupling with algebraic constraints is presented, e.g., in [18], where a stabilized index-2 co-simulation is used and both subsystems are driven by constraint forces.

Since we use an algebraic constraint to describe the coupling condition, no artificial stiffnesses or bushing elements are needed, cf. [10]. This is a substantial benefit of the strategy, because the numerical values of these artificial elements are in general unknown and, moreover, they influence significantly the stability behaviour of the coupled system, as shown in [5, 16].

We demonstrate the approach first with a small scale test example. With this test scenario, we analyze the accuracy and the stability of the approach and the influence of certain extrapolation schemes for the interface quantities. Second, we apply the approach to couple a realistic flexible structure, a geometrically exact Cosserat rod, cf. [14, 15], to a multibody system.

In particular, the paper at hand is organized as follows. In Sect. 2, the coupling approach is formulated for two subsystems and we show the equivalence of our approach to a monolithic formulation. After that, in Sect. 3, the chosen co-simulation method is discussed and in Sect. 4, we concentrate on the stability of the coupling approach together with the co-simulation scheme. Two applications of the approach will be shown in Sects. 5 and 6. First, the mentioned test example, a 2-mass spring-damper model, is used to analyze the convergence behaviour in detail. Then, to demonstrate the potential of our approach, we apply our coupling method to the more complex problem of a small multibody system with an embedded cable model, the mentioned geometrically exact Cosserat rod.

2 Force-displacement co-simulation with a kinematic coupling constraint

In this section, we show how the overall system can be decomposed into subsystems, which can be integrated separately without neglecting their dynamic interaction. To that end, the decomposed but kinematically coupled subsystems with system states x communicate over inputs u and outputs y. The exchanged quantities are forces and displacements, where the force-displacement coupling is an equivalent transformation of kinematically coupled subsystems, which are coupled via algebraic constraints.

For convenience, we assume only two coupled subsystems, and we name the subsystems flexible structure (F) and rigid multibody system (R). Here, the outputs y_R of the multibody system are system states, especially from the coupling interface, whereas the outputs y_F of the flexible structure are constraint forces. As sketched in Fig. 1, the coupling is realized as $u_R = y_F$ and $u_F = y_R$.

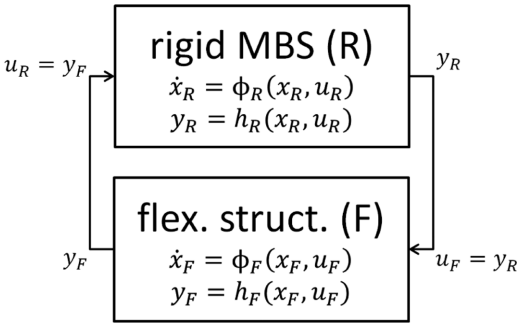

Fig. 1 Input-Output scheme of the decomposed subsystems: the multibody system with states x_R, inputs u_R, state equations $\dot{x}_R = \phi(x_R, u_R)$ and outputs $y_R = h_R(x_R, u_R)$ and the flexible structure with states x_F, inputs u_F, state equations $\dot{x}_F = \phi(x_F, u_F)$ and outputs $y_F = h_F(x_F, u_F)$.

To simulate the subsystems separately, i.e. for co-simulation, they exchange information at discrete points in time, the so-called macro points. Then, both subsystems integrate up to the next macro point independently from each other. For these

internal time steps, input quantities must be approximated from data at the macro points. More details on co-simulation are given in Sect. 3.

In our approach, the kinematic coupling is transformed to substitute a force-displacement co-simulation. Classically, this is achieved with an artificial stiffness, which couples the subsystems and significantly influences the stability of the co-simulation, cf. [5], and, moreover, the dynamic interaction might be significantly changed. Here, we use a kinematic coupling constraint and corresponding constraint forces. Thus, no artificial stiffness and no system modification are necessary.

2.1 Notation, kinematic coupling constraint, and monolithic equations

Let the rigid multibody system and the flexible structure with absolute coordinates $q_R(t) \in \mathbf{R}^{n_R}$, $q_F(t) \in \mathbf{R}^{n_F}$ and corresponding velocities $v_R(t) \in \mathbf{R}^{n_R}$, $v_F(t) \in \mathbf{R}^{n_F}$ be given as ordinary differential equations of motion

$$M_R \dot{v}_R(t) = f_R(q_R(t), v_R(t)) \quad \text{and} \quad M_F \dot{v}_F(t) = f_F(q_F(t), v_F(t)), \tag{1}$$

where the mass matrices M_R and M_F may depend on the states $q_R(t)$, resp. $q_F(t)$. To keep the notation simple, we assume

$$\dot{q}_R(t) = v_R(t) \qquad \text{and} \qquad \dot{q}_F(t) = v_F(t) \tag{2}$$

for the kinematic differential equations. Moreover, we only write q_R, q_F, etc. instead of $q_R(t)$, $q_F(t)$, etc.

To derive the coupling equations, we formulate the corresponding kinematic coupling constraints of the subsystems

$$0 = \hat{g}_{co}(q_R, q_F) \in \mathbf{R}^{n_{co}}, \tag{3}$$

where the n_{co} constraints are assumed to be independent. For readability reasons, these constraints are formulated scleronomically. An extension to rheonomic constraints is straightforward.

Due to the description with absolute coordinates, only a part of all the states in q_R and q_F are involved in the coupling, and we define them as $q_R^{co} \in \mathbf{R}^{n_{R,co}}$ and $q_F^{co} \in \mathbf{R}^{n_{F,co}}$. The remaining inner states $q_R^{in} \in \mathbf{R}^{n_{R,in}}$ and $q_F^{in} \in \mathbf{R}^{n_{F,in}}$ do not appear in the coupling constraints, and it holds

$$\frac{\partial \hat{g}_{co}}{\partial q_R^{co}} \neq 0, \quad \frac{\partial \hat{g}_{co}}{\partial q_F^{co}} \neq 0, \quad \frac{\partial \hat{g}_{co}}{\partial q_R^{in}} = 0, \quad \frac{\partial \hat{g}_{co}}{\partial q_F^{in}} = 0, \tag{4}$$

which leads to the slightly modified constraint formulation

$$0 = g_{co}(q_R^{co}, q_F^{co}). \tag{5}$$

Thus, with the partitioning

$$q_R = \begin{bmatrix} q_R^{co} \\ q_R^{in} \end{bmatrix}, \qquad q_F = \begin{bmatrix} q_F^{co} \\ q_F^{in} \end{bmatrix} \qquad \text{and, correspondingly,} \qquad v_R = \begin{bmatrix} v_R^{co} \\ v_R^{in} \end{bmatrix}, \qquad v_F = \begin{bmatrix} v_F^{co} \\ v_F^{in} \end{bmatrix} \tag{6}$$

the single subsystem equations without constraint forces read as

$$\begin{pmatrix} M_R^{co} & M_R^{ic} \\ M_R^{ci} & M_R^{in} \end{pmatrix} \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \end{bmatrix} = \begin{bmatrix} f_R^{co}(q_R, v_R) \\ f_R^{in}(q_R, v_R) \end{bmatrix} \quad \text{and} \quad \begin{pmatrix} M_F^{co} & M_F^{ic} \\ M_F^{ci} & M_F^{in} \end{pmatrix} \begin{bmatrix} \dot{v}_F^{co} \\ \dot{v}_F^{in} \end{bmatrix} = \begin{bmatrix} f_F^{co}(q_F, v_F) \\ f_F^{in}(q_F, v_F) \end{bmatrix}. \tag{7}$$

We assume that the constraints (5) are independent and that the constraint gradients

$$\frac{\partial g_{co}}{\partial (q_R, q_F)} \in \mathbf{R}^{n_{co} \times (n_R + n_F)} \tag{8}$$

have full rank n_{co}, and there exists at least one non-singular submatrix of $\frac{\partial g_{co}}{\partial (q_R, q_F)}$ of rank n_{co}, that defines a partitioning of all system states $q = [q_R^T, q_F^T]^T \in \mathbf{R}^{n_R + n_F}$ in dependent states and independent states, as shown in [20].

In the following, we assume that the states of the flexible structure q_F^{co} which are involved in the coupling are the dependent states, while the remaining states q_R^{co}, q_R^{in}, and q_F^{in} act as independent states. This means, we assume $q_F^{co} \in \mathbf{R}^{n_{co}}$ and $\frac{\partial g_{co}}{\partial q_F^{co}} \in \mathbf{R}^{n_{co} \times n_{co}}$ to be non-singular.

Differentiating the constraints (5) twice w.r.t. time results in the hidden constraints

$$\overset{\partial_t}{\Longrightarrow} 0 = \begin{pmatrix} \dfrac{\partial g_{co}}{\partial q_R^{co}} & 0 & \dfrac{\partial g_{co}}{\partial q_F^{co}} & 0 \end{pmatrix} \begin{bmatrix} v_R^{co} \\ v_R^{in} \\ v_F^{co} \\ v_F^{in} \end{bmatrix}, \tag{9}$$

$$\overset{\partial_t}{\Longrightarrow} 0 = \begin{pmatrix} \dfrac{\partial g_{co}}{\partial q_R^{co}} & 0 & \dfrac{\partial g_{co}}{\partial q_F^{co}} & 0 \end{pmatrix} \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \\ \dot{v}_F^{co} \\ \dot{v}_F^{in} \end{bmatrix} + g_{co}^{II}, \qquad g_{co}^{II} = \frac{\partial^2 g_{co}}{\partial q^2}(q)[v, v], \tag{10}$$

for the coupling velocities and accelerations.

With the algebraic constraints on acceleration level, we formulate the monolithic system

$$\left(\begin{array}{cc|cc|c} M_R^{co} & M_R^{ic} & & & \dfrac{\partial g_{co}}{\partial q_R^{co}}^T \\ M_R^{ci} & M_R^{in} & & & 0 \\ \hline & & M_F^{co} & M_F^{ic} & \dfrac{\partial g_{co}}{\partial q_F^{co}}^T \\ & & M_F^{ci} & M_F^{in} & 0 \\ \hline \dfrac{\partial g_{co}}{\partial q_R^{co}} & 0 & \dfrac{\partial g_{co}}{\partial q_F^{co}} & 0 & \end{array}\right) \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \\ \dot{v}_F^{co} \\ \dot{v}_F^{in} \\ \Lambda \end{bmatrix} = \begin{bmatrix} f_R^{co}(q_R, v_R) \\ f_R^{in}(q_R, v_R) \\ f_F^{co}(q_F, v_F) \\ f_F^{in}(q_F, v_F) \\ -g_{co}^{II} \end{bmatrix}, \tag{11}$$

which is of differentiation index 1, if the constraint Jacobian has full rank and the mass matrices are positive definite (see e.g. [12, Chap. VII.1, p.465]). The constraint forces with Lagrange multiplier Λ act on both subsystems.

If only the constraints on acceleration level are used as in (11), the numerical time-integration would lead to a drift-off in the kinematic constraints and its first time-derivative. To avoid this problem, projections on the constraint manifolds are required, see e.g. [10, 12].

Alternatively, starting from (11) we develop a method, where the coupling states of the flexible structure q_F^{co}, v_F^{co}, and $\dot{v}_F^{co}$ are prescribed from the multibody system and the kinematic coupling constraints. Thus, the flexible structure is not driven by forces anymore.

2.2 Derivation of the coupling equations

Theorem 2.1. *Let the subsystems* (7) *and the kinematic coupling constraints* (5) *with non-singular* $\frac{\partial g_{co}}{\partial q_F^{co}}$ *be given. Then solving the monolithic equations* (11) *is equivalent to solving the subsystems*

$$M_F^{in}\dot{v}_F^{in} = f_F^{in}(q_F^{in}, v_F^{in}, q_F^{co}, v_F^{co}) - M_F^{ci}\dot{v}_F^{co} =: \hat{f}_F^{in}(q_F^{in}, v_F^{in}, u_F) \tag{12}$$

and

$$\begin{pmatrix} M_R^{co} & M_R^{ic} \\ M_R^{ci} & M_R^{in} \end{pmatrix} \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \end{bmatrix} = \begin{bmatrix} f_R^{co}(q_R, v_R) \\ f_R^{in}(q_R, v_R) \end{bmatrix} - \begin{pmatrix} \dfrac{\partial g_{co}}{\partial q_R^{co}}^T \\ 0 \end{pmatrix} \Lambda =: \hat{f}_R(q_R, v_R, u_R), \tag{13}$$

where the constraint gradient is evaluated with $\frac{\partial g_{co}}{\partial q_R^{co}}(q_R^{co}, q_F^{co})|_{q_F^{co}=q_F^{co}(q_R^{co})}$*, meaning that* q_R^{co} *is sufficient to evaluate* $\frac{\partial g_{co}}{\partial q_R^{co}}$*, and the subsystem inputs*

$$u_F = y_R = \left[q_F^{co\,T},\ v_F^{co\,T},\ \dot{v}_F^{co\,T},\ q_R{}^T,\ v_R{}^T,\ \dot{v}_R{}^T \right]^T \tag{14}$$

and

$$u_R = y_F = \Lambda \tag{15}$$

are computed such that the coupling constraints (5) *and the hidden constraints* (9)*,* (10) *are fulfilled.*

Remark 2.2. Only the first three components of u_F are required in the equations of motion (12) and the last three seem to be unused. However, they might be necessary to compute the output $y_F = \Lambda$, as we will see later in this section.

To prove Theorem 2.1, we first show the following:

Lemma 2.3. *Let the coupling states* $q_R^{co} : \mathbf{R} \to \mathbf{R}^{n_{R,co}},\ t \mapsto q_R^{co}(t)$ *with* $\dot{q}_R^{co} = v_R^{co}$ *and* $q_F^{co} : \mathbf{R} \to \mathbf{R}^{n_{co}},\ t \mapsto q_F^{co}(t)$ *with* $\dot{q}_F^{co} = v_F^{co}$ *and the constraints* (5) *be given. Moreover, we assume that* $\frac{\partial g_{co}}{\partial q_F^{co}} \in \mathbf{R}^{n_{co} \times n_{co}}$ *is non-singular for all pairs* $(q_R^{co}, q_F^{co}) \in \mathbf{R}^{n_{R,co}} \times \mathbf{R}^{n_{co}}$.

Then, in the neighbourhood of every fixed but arbitrary $\hat{q}_R^{co}$, there exists a function ξ_q and even globally defined functions ξ_v and $\xi_{\dot{v}}$ with

$$\xi_q(q_R^{co}) = q_F^{co}, \quad \xi_v(q_R^{co}, v_R^{co}) = v_F^{co}, \quad \xi_{\dot{v}}(q_R^{co}, v_R^{co}, \dot{v}_R^{co}) = \dot{v}_F^{co}, \tag{16}$$

such that it holds

$$0 = g_{co}\left(q_R^{co}, \xi_q(q_R^{co})\right), \tag{17a}$$

$$0 = \frac{\partial g_{co}}{\partial q_R^{co}} v_R^{co} + \frac{\partial g_{co}}{\partial q_F^{co}} \xi_v(q_R^{co}, v_R^{co}), \tag{17b}$$

$$0 = \frac{\partial g_{co}}{\partial q_R^{co}} \dot{v}_R^{co} + \frac{\partial g_{co}}{\partial q_F^{co}} \xi_{\dot{v}}(q_R^{co}, v_R^{co}, \dot{v}_R^{co}) + g_{co}^{II}. \tag{17c}$$

P r o o f . In the neighbourhood of a fixed but arbitrary pair $(\hat{q}_R^{co}, \hat{q}_F^{co})$ which fulfills the constraints, we can apply the implicit function theorem to $0 = g_{co}(q_R^{co}, q_F^{co})$, leading locally to a unique function $\xi_q(q_R^{co}) = q_F^{co}$, such that it holds $0 = g_{co}(q_R^{co}, \xi_q(q_R^{co}))$ in a neighbourhood of $\hat{q}_R^{co}$.

Due to linearity, the hidden constraints immediately result in global expressions for the coupling velocities and accelerations of the flexible structure

$$\xi_v(q_R^{co}, v_R^{co}) = -\left(\frac{\partial g_{co}}{\partial q_F^{co}}\right)^{-1} \left(\frac{\partial g_{co}}{\partial q_R^{co}} v_R^{co}\right) = v_F^{co}, \tag{18}$$

$$\xi_{\dot{v}}(q_R^{co}, v_R^{co}, \dot{v}_R^{co}) = -\left(\frac{\partial g_{co}}{\partial q_F^{co}}\right)^{-1} \left(\frac{\partial g_{co}}{\partial q_R^{co}} \dot{v}_R^{co} + g_{co}^{II}\right) = \dot{v}_F^{co}. \tag{19}$$

□

Remark 2.4. The derivation of v_F^{co} and $\dot{v}_F^{co}$ is similar to the coordinate partitioning in [20], with dependent states q_F^{co} and independent states $[q_R^{coT}, q_R^{inT}, q_F^{inT}]^T$. Additionally, q_F^{co} is computed via the kinematic constraints. In general, this has to be done numerically and leads to some extra effort. However, the common kinematic coupling constraints often allow an explicit expression of q_F^{co}. Finally, we get prescribed coupling states q_F^{co}, v_F^{co}, and $\dot{v}_F^{co}$ for the flexible structure, which no longer have to be solved by their equations of motion.

Now, we prove Theorem 2.1:

P r o o f . We write the monolithic equations as follows

$$\begin{pmatrix} M_R^{co} & M_R^{ic} \\ M_R^{ci} & M_R^{in} \end{pmatrix} \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \end{bmatrix} = \begin{bmatrix} f_R^{co}(q_R, v_R) \\ f_R^{in}(q_R, v_R) \end{bmatrix} - \begin{pmatrix} \frac{\partial g_{co}}{\partial q_R^{co}}(q_R^{co}, q_F^{co})^T \\ 0 \end{pmatrix} \Lambda, \tag{20a}$$

$$\begin{pmatrix} M_F^{co} & M_F^{ic} \\ M_F^{ci} & M_F^{in} \end{pmatrix} \begin{bmatrix} \dot{v}_F^{co} \\ \dot{v}_F^{in} \end{bmatrix} = \begin{bmatrix} f_F^{co}(q_F, v_F) \\ f_F^{in}(q_F, v_F) \end{bmatrix} - \begin{pmatrix} \frac{\partial g_{co}}{\partial q_F^{co}}(q_R^{co}, q_F^{co})^T \\ 0 \end{pmatrix} \Lambda, \tag{20b}$$

$$0 = g_{co}(q_R^{co}, q_F^{co}), \tag{20c}$$

where both subsystems are driven by constraint forces.

Applying Lemma 2.3, the prescribed coupling states q_F^{co}, v_F^{co}, and $\dot{v}_F^{co}$ can be computed from q_R^{co}, v_R^{co}, and $\dot{v}_R^{co}$. Therefore, we can skip the corresponding equations of motion in the flexible structure, instead of (20b) only considering

$$M_F^{in} \dot{v}_F^{in} = f_F^{in}(q_F^{in}, v_F^{in}, q_F^{co}, v_F^{co}) - M_F^{ci} \dot{v}_F^{co} =: \hat{f}_F^{in}\left(q_F^{in}, v_F^{in}, u_F\right), \tag{21}$$

and get rid of the constraint forces for the flexible structure. The constraint forces acting on the multibody system are interpreted as inputs $u_R = \Lambda$ of the multibody system and are the same as in the monolithic formulation. The corresponding constraint gradient $\frac{\partial g_{co}}{\partial q_R^{co}}(q_R^{co}, q_F^{co})$ can be evaluated only from q_R^{co}, since we can compute q_F^{co} from q_R^{co}. Thus, we can define the right-hand side as $\hat{f}_R(q_R, v_R, u_R)$. □

The evaluation of Λ still needs to be discussed. A straightforward computation would be to solve the linear monolithic index-1-system (11) to get Λ. However, if the subsystems become large, this is computationally expensive. Alternatively, the constraint forces can be derived from a much smaller system, exploiting the zero-blocks in the constraint Jacobian. For

known inner accelerations $\dot{v}_R^{in}$ and $\dot{v}_F^{in}$, we can get Λ from the first block rows of (20a) and (20b) combined with the hidden constraints (10), resulting in the smaller system of linear equations

$$\left(\begin{array}{c|c|c} M_R^{co} & & \frac{\partial g_{co}}{\partial q_R^{co}}^T \\ \hline & M_F^{co} & \frac{\partial g_{co}}{\partial q_F^{co}}^T \\ \hline \frac{\partial g_{co}}{\partial q_R^{co}} & \frac{\partial g_{co}}{\partial q_F^{co}} & \end{array}\right) \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_F^{co} \\ \Lambda \end{bmatrix} = \begin{bmatrix} f_R^{co}(q_R, v_R) - M_R^{ic}\dot{v}_R^{in} \\ f_F^{co}(q_F, v_F) - M_F^{ic}\dot{v}_F^{in} \\ -g_{co}^{II} \end{bmatrix}. \tag{22}$$

The dimension of the resulting system relates to the number of coupling states.

One step further, we could only use the middle block row of (22) to compute Λ leading to

$$\Lambda = \left(\frac{\partial g_{co}}{\partial q_F^{co}}\right)^{-1} \left(f_F^{co}(q_F, v_F) - M_F^{co}\dot{v}_F^{co} - M_F^{ic}\dot{v}_F^{in}\right). \tag{23}$$

Here, the entire acceleration vector $\dot{v}_F = \left[\, \dot{v}_F^{coT} \;\; \dot{v}_F^{inT} \,\right]^T$ of the flexible structure is required. This evaluation of Λ was also proposed in [20].

3 Co-simulation scheme for the coupling approach

In this paper, we use a co-simulation scheme of Jacobi type (see e.g. [4, 17]), i.e. the subsystems are integrated separately and in parallel. The co-simulation proceeds in macro steps $t_k \to t_{k+1} = t_k + h$ with macro step size h, where inside of each macro step the subsystems are integrated independently using approximated inputs. Due to the parallel integration, these approximations are defined with data from previous macro points and, thus, the inputs of both subsystems are extrapolated. Alternatively, a co-simulation of Gauss-Seidel type with a sequential integration could be applied. Here, the subsystem that integrates first uses extrapolated inputs, whereas the subsystem integrating hereafter can use interpolated inputs. The co-simulation schemes and the sequence of computation are discussed, e.g., in [4] for applications from electrical engineering. However, in this work, we focus on parallel co-simulation, because this scheme is more efficient than the sequential one.

3.1 Parallel integration and extrapolated subsystem inputs

At each macro point t_k from $t = t_0$ to the end point $t = t_{end}$, the subsystems provide outputs $y_{R,k}$ and $y_{F,k}$ which then give inputs $u_{R,k} = y_{F,k}$ and $u_{F,k} = y_{R,k}$. These inputs are used together with inputs from previous macro points to define an interpolation polynomial and to approximate $u_R(t)$ and $u_F(t)$ for all $t \in [t_k, t_{k+1}]$, i.e. we extrapolate the inputs, as depicted in Fig. 2.

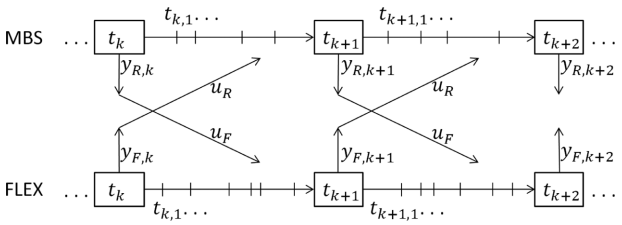

Fig. 2 Parallel co-simulation (Jacobi type) with extrapolated inputs.

To initialize the co-simulation, at t_0 only constant extrapolation is possible due to the lack of previous macro points. In the following macro steps, one successively increases the order of the interpolation polynomial, until the desired order is reached.

The inputs for the flexible structure are prescribed coupling states and the multibody system states evaluated at time t_k

$$u_{F,k} := u_F(t_k). \tag{24}$$

In the subsequent micro time steps up to t_{k+1}, coupling states from the past

$$u_{F,k-i} \;\text{ for }\; i = 0, ..., s \tag{25}$$

are used to extrapolate u_F in $[t_k, t_{k+1}]$ by the interpolation polynomial of order s with data $(t_{k-i}, u_{F,k-i})$ for $i = 0, ..., s$ and we call this function $u_{F,k}^{\text{extrap},s}$.

Remark 3.1. The components of the input u_F require special attention regarding consistency. It holds

$$u_F = \begin{bmatrix} q_F^{co} \\ v_F^{co} \\ \dot{v}_F^{co} \\ q_R \\ v_R \\ \dot{v}_R \end{bmatrix} = \begin{bmatrix} q_F^{co} \\ \dot{q}_F^{co} \\ \xi_{\dot{v}}(q_R^{co}, v_R^{co}, \dot{v}_R^{co}) \\ q_R \\ \dot{q}_R \\ M_R^{-1} \hat{f}_R(q_R, v_R, u_R) \end{bmatrix} \quad \text{with} \quad \dot{v}_R^{co} \quad \text{from} \quad \dot{v}_R = \begin{bmatrix} \dot{v}_R^{co} \\ \dot{v}_R^{in} \end{bmatrix} \tag{26}$$

and the extrapolation scheme for velocities $v_F^{co} = \dot{q}_F^{co}$ and $v_R = \dot{q}_R$ must be consistent with the one for positions q_F^{co} and q_R. For instance, a linear extrapolation for positions leads to a constant velocity extrapolation. However, the acceleration $\dot{v}_F^{co}$ and $\dot{v}_R$ are right hand side evaluations and are extrapolated with the same scheme as the positions.

The multibody system input, namely the constraint forces

$$u_{R,k} := u_R(t_k) \tag{27}$$

are computed at time t_k. Again, in the subsequent micro time steps up to t_{k+1}, previous constraint forces

$$u_{R,k-i} \quad \text{for} \quad i = 0, ..., r \tag{28}$$

are used to extrapolate u_R in $[t_k, t_{k+1}]$ by the interpolation polynomial of order r with data $(t_{k-i}, u_{R,k-i})$ for $i = 0, ..., r$ and we call this function $u_{R,k}^{\text{extrap},r}$.

3.2 Extrapolation formulas

Assuming inputs $u_k,\ u_{k-1},\ \ldots,\ u_{k-r} \in \mathbf{R}^m$ at previous macro points $t_k,\ t_{k-1},\ \ldots,\ t_{k-r}$ are given and we construct the interpolation polynomial $u_k^{\text{extrap},r}$ of order r with this data. Then the polynomial can be written as

$$u_k^{\text{extrap},r}(t) = \mathcal{T}_m^r(t, t_k, \ldots, t_{k-r}) \begin{bmatrix} u_k \\ u_{k-1} \\ \vdots \\ u_{k-r} \end{bmatrix} \tag{29}$$

and the evaluation of the polynomial at time $t = t_k + \tau$ with $\tau \geq 0$, i.e. the extrapolation, for equidistant macro step sizes $h = t_k - t_{k-1}$ for all k, leads to the expressions

$$\begin{aligned} \mathcal{T}_m^0(\tau, h) &= \mathbf{I}_m, \qquad \mathcal{T}_m^1(\tau, h) = \left(\left(1 + \frac{\tau}{h}\right)\mathbf{I}_m \left(-\frac{\tau}{h}\right)\mathbf{I}_m\right), \\ \mathcal{T}_m^2(\tau, h) &= \left(\left(1 + \frac{3\tau}{2h} + \frac{\tau^2}{2h^2}\right)\mathbf{I}_m \quad \left(-\frac{4\tau}{2h} - \frac{2\tau^2}{2h^2}\right)\mathbf{I}_m \quad \left(\frac{\tau}{2h} + \frac{\tau^2}{2h^2}\right)\mathbf{I}_m\right), \end{aligned} \tag{30}$$

for constant ($r = 0$), linear ($r = 1$), and quadratic ($r = 2$) extrapolation. These classical formulas are used in the following section and, therefore, are stated here explicitly.

4 Stability of the coupling approach

In this section, we analyze the stability of the presented coupling approach. Therefore, we check our approach for upcoming algebraic loops and their behaviour during time integration.

One important aspect is zero-stability, which guarantees a stable simulation for sufficiently small time steps. To analyze zero-stability, we proceed as in [13] and consider loops in the input-output scheme of the proposed co-simulation approach.

Moreover, we want to have a look on the stability of the simulation for finite time step sizes. To that end, we analyze states from subsequent time steps and their iterative behaviour.

4.1 Analysis of zero-stability

A well-known loop is the one that appears due to feed-through configurations in the subsystems and, consequently, zero-stability can not be guaranteed anymore, cf. [13]. If an efficient computation of the constraint forces is used, i.e. the smaller

linear system (22) or the skipped equations of motion (23) are used to compute Λ, one has to deal with this feed-through. The reason for this are accelerations which appear on the right hand side in Eqs. (22) or (23) and can only be evaluated with extrapolated inputs from previous time-steps, leading to approximated forces.

As long as the full monolithic system (11) is used for the exact computation of the constraint forces, no feed-through occurs and zero-stability is given. This is because only displacements and velocities are required here.

In general, for the inputs and outputs, it holds

$$u_F = y_R = \left[{q_F^{co}}^T, {v_F^{co}}^T, {\dot{v}_F^{co}}^T, {q_R}^T, {v_R}^T, {\dot{v}_R}^T \right]^T \in \mathbf{R}^{3(n_{co}+n_R)} \quad \text{and} \quad u_R = y_F = \Lambda \in \mathbf{R}^{n_{co}} \tag{31}$$

and we can write

$$\begin{bmatrix} u_R \\ u_F \end{bmatrix} = \underbrace{\begin{pmatrix} 0 & \mathbf{I}_{n_{co}} \\ \mathbf{I}_{3(n_{co}+n_R)} & 0 \end{pmatrix}}_{=:L} \begin{bmatrix} y_R \\ y_F \end{bmatrix}. \tag{32}$$

Moreover, the output y_R is always given as

$$y_R = \begin{bmatrix} q_F^{co} \\ v_F^{co} \\ \dot{v}_F^{co} \\ q_R \\ v_R \\ \dot{v}_R \end{bmatrix} = \begin{bmatrix} \xi_q(q_R^{co}) \\ \xi_v(q_R^{co}, v_R^{co}) \\ \xi_{\dot{v}}(q_R^{co}, v_R^{co}, \dot{v}_R^{co}) \\ q_R \\ v_R \\ M_R^{-1} \hat{f}_R(q_R, v_R, u_R) \end{bmatrix} =: h_R(q_R, v_R, u_R) \tag{33a}$$

$$\approx h_R(\bar{q}_R, \bar{v}_R, \bar{u}_R) + \underbrace{\frac{\partial h_R}{\partial (q_R, v_R)}}_{=:C_R} \left(\begin{bmatrix} q_R \\ v_R \end{bmatrix} - \begin{bmatrix} \bar{q}_R \\ \bar{v}_R \end{bmatrix} \right) + \underbrace{\frac{\partial h_R}{\partial u_R}}_{=:D_R} (u_R - \bar{u}_R) \tag{33b}$$

whereas the evaluation of $y_F = \Lambda$ differs depending on the chosen approach and we write in general

$$y_F =: h_F(q_F^{in}, v_F^{in}, u_F) \approx h_F(\bar{q}_F^{in}, \bar{v}_F^{in}, \bar{u}_F) + \underbrace{\frac{\partial h_F}{\partial (q_F^{in}, v_F^{in})}}_{=:C_F} \left(\begin{bmatrix} q_F^{in} \\ v_F^{in} \end{bmatrix} - \begin{bmatrix} \bar{q}_F^{in} \\ \bar{v}_F^{in} \end{bmatrix} \right) + \underbrace{\frac{\partial h_F}{\partial u_F}}_{=:D_F} (u_F - \bar{u}_F). \tag{34}$$

Moreover, for the partial derivative w.r.t. u_F, it holds $D_F = \frac{\partial h_F}{\partial u_F} = \begin{pmatrix} \frac{\partial h_F}{\partial q_F^{co}} & \frac{\partial h_F}{\partial v_F^{co}} & \frac{\partial h_F}{\partial \dot{v}_F^{co}} & \frac{\partial h_F}{\partial q_R} & \frac{\partial h_F}{\partial v_R} & \frac{\partial h_F}{\partial \dot{v}_R} \end{pmatrix}$.

The above linearizations with Jacobian evaluations, for instance, in equilibrium points $(\bar{q}_R, \bar{v}_R, \bar{u}_R)$ and $(\bar{q}_F^{in}, \bar{v}_F^{in}, \bar{u}_F)$, are used to formulate the loop of the outputs $\begin{bmatrix} \Delta y_R \\ \Delta y_F \end{bmatrix} := \begin{bmatrix} y_R \\ y_F \end{bmatrix} - \begin{bmatrix} h_R(\bar{q}_R, \bar{v}_R, \bar{u}_R) \\ h_F(\bar{q}_F^{in}, \bar{v}_F^{in}, \bar{u}_F) \end{bmatrix}$ as

$$\begin{bmatrix} \Delta y_R \\ \Delta y_F \end{bmatrix} \approx \underbrace{\begin{pmatrix} C_R & \\ & C_F \end{pmatrix}}_{=:C} \begin{bmatrix} \Delta x_R \\ \Delta x_F \end{bmatrix} + \underbrace{\begin{pmatrix} D_R & \\ & D_F \end{pmatrix}}_{=:D} \begin{bmatrix} \Delta u_R \\ \Delta u_F \end{bmatrix} \overset{u=Ly}{=} C \begin{bmatrix} \Delta x_R \\ \Delta x_F \end{bmatrix} + DL \begin{bmatrix} \Delta y_R \\ \Delta y_F \end{bmatrix} \tag{35}$$

with states $x_R = [q_R^T, v_R^T]^T$ and $x_F = [q_F^{inT}, v_F^{inT}]^T$. Consequently, the stability analysis is only reliable close to this equilibrium. Often, this is sufficient. However, to improve the results, linearization could be performed for several points and the closest one is taken into consideration.

As in [13], for zero-stability we have to check, if the spectral radius of DL fulfills $\rho(DL) < 1$. For the characteristic polynomial of DL, we get with $n_{RF} = 4n_{co} + 3n_R$

$$\begin{aligned} \det(\mu \mathbf{I}_{n_{RF}} - DL) &= \det \left(\begin{array}{c|c} \mu \mathbf{I}_{3(n_{co}+n_R)} & -D_R \\ \hline -D_F & \mu \mathbf{I}_{n_{co}} \end{array} \right) \overset{\mu \neq 0}{=} \det\left(\mu \mathbf{I}_{3(n_{co}+n_R)}\right) \det\left(\mu \mathbf{I}_{n_{co}} - \frac{1}{\mu} D_F D_R\right) \\ &= \mu^{3(n_{co}+n_R)} \det\left(\mu \mathbf{I}_{n_{co}} - \frac{1}{\mu} \left(\frac{\partial h_F}{\partial \dot{v}_F^{co}} \frac{\partial g_{co}}{\partial q_F^{co}}^{-1} \frac{\partial g_{co}}{\partial q_R^{co}} \begin{pmatrix} \mathbf{I}_{n_{R,co}} & 0 \end{pmatrix} - \frac{\partial h_F}{\partial \dot{v}_R} \right) M_R^{-1} \frac{\partial g_{co}}{\partial q_R}^T \right). \end{aligned} \tag{36}$$

The last equality is given by multiplying the general $D_F = \begin{pmatrix} \frac{\partial h_F}{\partial q_F^{co}} & \frac{\partial h_F}{\partial v_F^{co}} & \frac{\partial h_F}{\partial \dot{v}_F^{co}} & \frac{\partial h_F}{\partial q_R} & \frac{\partial h_F}{\partial v_R} & \frac{\partial h_F}{\partial \dot{v}_R} \end{pmatrix}$ with

$$D_R = \frac{\partial h_R}{\partial u_R} = \frac{\partial h_R}{\partial \Lambda} = \begin{pmatrix} 0 \\ 0 \\ \left(\frac{\partial g_{co}}{\partial q_F^{co}}\right)^{-1} \frac{\partial \boldsymbol{g}_{co}}{\partial \boldsymbol{q}_R^{co}} \begin{pmatrix} \mathbf{I}_{n_{R,co}} & 0 \end{pmatrix} \boldsymbol{M}_R^{-1} \frac{\partial g_{co}}{\partial q_R}^T \\ 0 \\ 0 \\ -\boldsymbol{M}_R^{-1} \frac{\partial g_{co}}{\partial q_R}^T \end{pmatrix}, \tag{37}$$

where (13) together with (19) is inserted in (33a).

4.1.1 Exact force computation with full monolithic system

For an evaluation of Λ with system (11), position and velocity states q_R, v_R, q_F, and v_F are required and we can write

$$y_F = \Lambda(q_F, v_F, q_R, v_R) = h_F(q_F^{in}, v_F^{in}, u_F), \tag{38}$$

where $u_F^{(3)} = \dot{v}_F^{co}$ and $u_F^{(6)} = \dot{v}_R$ are unused. Here, zero-stability is given, because it holds

$$\frac{\partial h_F}{\partial \dot{v}_F^{co}} = 0, \quad \frac{\partial h_F}{\partial \dot{v}_R} = 0 \qquad \Longrightarrow \qquad \det\left(\mu \mathbf{I}_{n_{RF}} - DL\right) \overset{\mu \neq 0}{=} \mu^{3(n_{co}+n_R)} \det\left(\mu \mathbf{I}_{n_{co}}\right) \tag{39}$$

and $0 = \det(\mu \mathbf{I}_{n_{RF}} - DL)$ is only true for eigenvalues $\mu = 0$.

4.1.2 Efficient force approximation

If the subsystems become large, it is reasonable to approximate the constraint forces to decrease the computation time. If we use the smaller linear system (22) to evaluate Λ, then - besides positions and velocities - inner accelerations are required for the constraint force computation and it holds

$$y_F = \Lambda(q_F, v_F, \dot{v}_F^{in}, q_R, v_R, \dot{v}_R^{in}) \qquad \text{with} \qquad \dot{v}_F^{in} = \left(M_F^{in}\right)^{-1} \left(f_F^{in}(q_F, v_F) - M_F^{ci} \dot{v}_F^{co}\right) \tag{40a}$$

$$= \tilde{h}_F(q_F^{in}, v_F^{in}, u_F). \tag{40b}$$

Here we get $\frac{\partial \tilde{h}_F}{\partial \dot{v}_F^{co}} \neq 0$ and $\frac{\partial \tilde{h}_F}{\partial \dot{v}_R} \neq 0$ and, in general, we have non-zero eigenvalues in the spectrum of DL. In detail, it holds

$$\frac{\partial \tilde{h}_F}{\partial \dot{v}_F^{co}} = \left(\frac{\partial g_{co}}{\partial q_R^{co}} \left(M_R^{co}\right)^{-1} \frac{\partial g_{co}}{\partial q_R^{co}}^T + \frac{\partial g_{co}}{\partial q_F^{co}} \left(M_F^{co}\right)^{-1} \frac{\partial g_{co}}{\partial q_F^{co}}^T \right)^{-1} \frac{\partial g_{co}}{\partial q_F^{co}} \left(M_F^{co}\right)^{-1} M_F^{ic} \left(M_F^{in}\right)^{-1} M_F^{ci} \tag{41a}$$

$$\frac{\partial \tilde{h}_F}{\partial \dot{v}_R} = \left(\frac{\partial g_{co}}{\partial q_R^{co}} \left(M_R^{co}\right)^{-1} \frac{\partial g_{co}}{\partial q_R^{co}}^T + \frac{\partial g_{co}}{\partial q_F^{co}} \left(M_F^{co}\right)^{-1} \frac{\partial g_{co}}{\partial q_F^{co}}^T \right)^{-1} \left(0 \middle| -\frac{\partial g_{co}}{\partial q_R^{co}} \left(M_R^{co}\right)^{-1} M_R^{ic} \right) \tag{41b}$$

and if the off-diagonal blocks of the mass matrices are zero, i.e. $M_F^{ic} = 0$ and $M_R^{ic} = 0$, then also the Jacobian $\frac{\partial \tilde{h}_F}{\partial \dot{v}_F^{co}}$ and $\frac{\partial \tilde{h}_F}{\partial \dot{v}_R}$ vanish and zero-stability is guaranteed again.

A similar observation results from the approximation of Λ with (23), where we get

$$y_F = \Lambda(q_F, v_F, \dot{v}_F^{co}, \dot{v}_F^{in}) = \tilde{\tilde{h}}_F(q_F^{in}, v_F^{in}, u_F) \tag{42}$$

and it holds

$$\frac{\partial \tilde{\tilde{h}}_F}{\partial \dot{v}_F^{co}} = \left(\frac{\partial g_{co}}{\partial q_F^{co}}^T \right)^{-1} \left(-M_F^{co} + M_F^{ic} \left(M_F^{in}\right)^{-1} M_F^{ci} \right), \qquad \frac{\partial \tilde{\tilde{h}}_F}{\partial \dot{v}_R} = 0. \tag{43}$$

But here, even with vanishing off-diagonal blocks of the mass matrices, we have a non-zero Jacobian $\frac{\partial \tilde{\tilde{h}}_F}{\partial \dot{v}_F^{co}}$ and can not guarantee zero-stability.

In general, from equations (36), (41a), (41b), and (43), it is obvious that the subsystem masses are decisive for the stability of the co-simulation. A similar observation was made in [3]. The mass-ratio of the subsystems, resp. the ordering of the subsystems decides on the stability of the co-simulation.

4.2 Stability for finite time-steps

Besides the zero-stability, i.e. a stable co-simulation for macro steps $h \longrightarrow 0$, we want to analyze the stability for finite macro steps. In this context, it is reasonable to assume that the mechanical subsystems itself are stable or asymptotically stable (dissipative systems).

We start at time t_k and consider the macro step with step size h up to $t_{k+1} = t_k + h$ for the linearized systems

$$\dot{x}_R = A_R x_R + B_R u_R, \qquad \dot{x}_F = A_F x_F + B_F u_F \tag{44}$$

with states $x_R = \left[\begin{smallmatrix} q_R \\ v_R \end{smallmatrix}\right]$, $x_F = \left[\begin{smallmatrix} q_F^{in} \\ v_F^{in} \end{smallmatrix}\right]$ and Jacobians evaluated in, for instance, equilibrium points $(\bar{x}_R, \bar{u}_R)$ and $(\bar{x}_F, \bar{u}_F)$ given as

$$A_R = \begin{pmatrix} 0 & \mathbf{I}_{n_R} \\ \dfrac{\partial \left(M_R^{-1} \hat{f}_R\right)}{\partial q_R} & \dfrac{\partial \left(M_R^{-1} \hat{f}_R\right)}{\partial v_R} \end{pmatrix}, \qquad B_R = \begin{pmatrix} 0 \\ -M_R^{-1} \dfrac{\partial g_{co}}{\partial q_R}^T \end{pmatrix}, \tag{45a}$$

$$A_F = \begin{pmatrix} 0 & \mathbf{I}_{n_F - n_{co}} \\ \dfrac{\partial \left(M_F^{in\,-1}\, \hat{f}_F^{in}\right)}{\partial q_R^{in}} & \dfrac{\partial \left(M_F^{in\,-1}\, \hat{f}_F^{in}\right)}{\partial v_R^{in}} \end{pmatrix}, \qquad B_F = \begin{pmatrix} 0 \\ \dfrac{\partial \left(M_F^{in\,-1}\, \hat{f}_F^{in}\right)}{\partial u_F} \end{pmatrix}. \tag{45b}$$

(Here, and also in the previous linearization, the linearization points fulfill the coupling constraints in any case, since only inner states of the flexible structure are considered.)

Moreover, we define $x = \left[\begin{smallmatrix} x_R \\ x_F \end{smallmatrix}\right] \in \mathbf{R}^{n_x}$, $y = \left[\begin{smallmatrix} y_R \\ y_F \end{smallmatrix}\right] \in \mathbf{R}^{n_y}$, $u = \left[\begin{smallmatrix} u_R \\ u_F \end{smallmatrix}\right] \in \mathbf{R}^{n_u}$ (with $n_y = n_u$) and matrices

$$A = \begin{pmatrix} A_R & \\ & A_F \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} B_R & \\ & B_F \end{pmatrix}, \tag{46}$$

such that for subsequent time steps $x_k = x(t_k)$ and $x_{k+1} = x(t_{k+1}) \in \mathbf{R}^n$ with step size $h = t_{k+1} - t_k$ we get

$$x_{k+1} = \mathrm{e}^{Ah} x_k + \int_0^h \mathrm{e}^{A(h-\tau)} B u(t_k + \tau) d\tau, \tag{47a}$$

$$y_{k+1} = C x_{k+1} + D u_{k+1}. \tag{47b}$$

During the step from t_k to t_{k+1} only extrapolated inputs can be used and we write

$$\begin{aligned} u(t_k + \tau) \approx \mathcal{T}_{n_u}^r(\tau, h) \begin{bmatrix} u_k \\ \vdots \\ u_{k-r} \end{bmatrix} &= \mathcal{T}_{n_u}^r(\tau, h) \begin{bmatrix} L\left(C x_k + D L y_k\right) \\ \vdots \\ L\left(C x_{k-r} + D L y_{k-r}\right) \end{bmatrix} \\ &= LC\mathcal{T}_{n_x}^r(\tau, h) \begin{bmatrix} x_k \\ \vdots \\ x_{k-r} \end{bmatrix} + LDL\mathcal{T}_{n_y}^r(\tau, h) \begin{bmatrix} y_k \\ \vdots \\ y_{k-r} \end{bmatrix}. \end{aligned} \tag{48}$$

Moreover, for $u_{k+1} = u(t_k + h)$ in the output y_{k+1} we write

$$u(t_k + h) \approx \mathcal{T}_{n_u}^r(h, h) \begin{bmatrix} u_k \\ \vdots \\ u_{k-r} \end{bmatrix} = L\mathcal{T}_{n_y}^r(h, h) \begin{bmatrix} y_k \\ \vdots \\ y_{k-r} \end{bmatrix}, \tag{49}$$

the macro step size h cancels in $\mathcal{T}_{n_y}^r(h,h)$ and for constant, linear and quadratic extrapolation it holds

$$\mathcal{T}_{n_y}^0(h,h) = \mathbf{I}_{n_y}, \qquad \mathcal{T}_{n_y}^1(h,h) = \begin{pmatrix} 2\mathbf{I}_{n_y} & -\mathbf{I}_{n_y} \end{pmatrix}, \qquad \mathcal{T}_{n_y}^2(h,h) = \begin{pmatrix} 3\mathbf{I}_{n_y} & -3\mathbf{I}_{n_y} & \mathbf{I}_{n_y} \end{pmatrix}. \tag{50}$$

Summarizing, this results in the following iteration for subsequent time-steps

$$\begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+1-r} \\ y_{k+1} \\ \vdots \\ y_{k+1-r} \end{bmatrix} = \underbrace{\left(\begin{array}{ccc|ccc} & \mathcal{S}_{xx}(h) & & & \mathcal{S}_{xy}(h) & \\ \mathbf{I}_{n_x} & & 0 & 0 & \cdots & 0 \\ & \ddots & \vdots & \vdots & & \vdots \\ & & \mathbf{I}_{n_x}\ 0 & 0 & \cdots & 0 \\ \hline & \mathcal{S}_{yx}(h) & & & \mathcal{S}_{yy}(h) & \\ 0 & \cdots & 0 & \mathbf{I}_{n_y} & & 0 \\ \vdots & & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & & & \mathbf{I}_{n_y}\ 0 \end{array}\right)}_{=:\mathcal{S}(h)} \begin{bmatrix} x_k \\ \vdots \\ x_{k-r} \\ y_k \\ \vdots \\ y_{k-r} \end{bmatrix}, \tag{51}$$

with matrices

$$\mathcal{S}_{xx}(h) = \begin{pmatrix} \mathrm{e}^{Ah} & 0 & \cdots & 0 \end{pmatrix} + \int_0^h \mathrm{e}^{A(h-\tau)} BLC\mathcal{T}_{n_x}^r(\tau,h)d\tau, \tag{52a}$$

$$\mathcal{S}_{xy}(h) = \int_0^h \mathrm{e}^{A(h-\tau)} BLDL\mathcal{T}_{n_y}^r(\tau,h)d\tau, \tag{52b}$$

$$\mathcal{S}_{yx}(h) = \begin{pmatrix} C\mathrm{e}^{Ah} & 0 & \cdots & 0 \end{pmatrix} + C\int_0^h \mathrm{e}^{A(h-\tau)} BLC\mathcal{T}_{n_x}^r(\tau,h)d\tau = C\mathcal{S}_{xx}(h), \tag{52c}$$

$$\mathcal{S}_{yy}(h) = C\int_0^h \mathrm{e}^{A(h-\tau)} BLDL\mathcal{T}_{n_y}^r(\tau,h)d\tau + DL\mathcal{T}_{n_y}^r(h,h) = C\mathcal{S}_{xy}(h) + DL\mathcal{T}_{n_y}^r(h,h). \tag{52d}$$

In long-term simulation, a stable co-simulation can only be guaranteed, if the spectral radius of $\mathcal{S}(h)$ fulfills

$$\rho(\mathcal{S}(h)) < 1. \tag{53}$$

For numerical simulation, this is useful to find suitable macro step sizes h and extrapolation schemes. In Sects. 5 and 6, we use these investigations to test the example systems for stability.

Remark 4.1. The lower right block of $\mathcal{S}(h)$ contains the term $DL\mathcal{T}_{n_y}^r(h,h)$, which is independent of the step size. We have already seen it in the analysis of zero-stability, but without the extrapolation term $\mathcal{T}_{n_y}^r(h,h)$. Only for constant extrapolation it holds $DL\mathcal{T}_{n_y}^r(h,h) = DL$.

To include the extrapolation scheme in the zero-stability analysis of the previous section, we compute the spectral radius of the second term in the lower right block, i.e. the lower right block of $\mathcal{S}(h)$ with $h \longrightarrow 0$. Therefore, we consider the characteristic polynomial

$$\det\left(\mu\mathbf{I}_{(r+1)n_y} - \begin{pmatrix} DL\mathcal{T}_{n_y}^r(h,h) & \\ \mathbf{I}_{n_y} & 0 \\ & \ddots\ \vdots \\ & \mathbf{I}_{n_y}\ 0 \end{pmatrix}\right) \overset{\mu\neq 0}{=} \mu^{rn_y} \cdot \det\left(\mu\mathbf{I}_{n_y} - \left(\sum_{i=0}^r \frac{\alpha_i}{\mu^i}\right) DL\right)$$

$$= \mu^{\beta} \det\left(\mu\mathbf{I}_{n_{co}} - \frac{1}{\mu}\left(\sum_{i=0}^r \frac{\alpha_i}{\mu^i}\right)^2 \left(\frac{\partial h_F}{\partial \dot{v}_F^{co}} \frac{\partial g_{co}}{\partial q_F^{co}}^{-1} \frac{\partial g_{co}}{\partial q_R^{co}} (\mathbf{I}_{n_{R,co}} \quad 0) - \frac{\partial h_F}{\partial \dot{v}_R}\right) M_R^{-1} \frac{\partial g_{co}}{\partial q_R}^T\right), \tag{54}$$

where $\beta = rn_y + 3(n_{co} + n_R)$ and the α_i, with $i = 0, 1$, are the constants in $\mathcal{T}_{n_y}^r(h,h) = (\alpha_0\mathbf{I}_{n_y} \quad \cdots \quad \alpha_r\mathbf{I}_{n_y})$. This looks similar to (36), except of the constants from extrapolation, which now appear. Thus, we can include the influence of the extrapolation scheme in the analysis of zero-stability.

5 Test problem: 2-mass spring-damper model

To test the presented co-simulation method concerning stability and convergence, we set up a simple model for numerical experiments. The analytical solution of that model is known and, thus, allows a detailed convergence analysis.

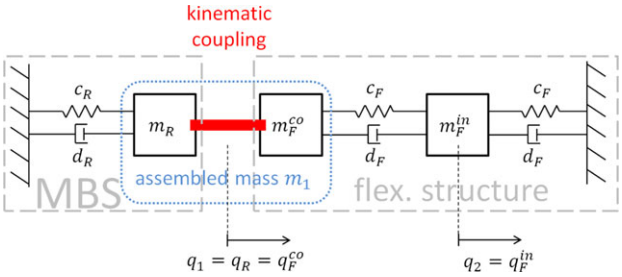

Fig. 3 Test model: 2-mass spring-damper.

5.1 The test model and its analytical solution

We consider a 2-mass spring-damper model with two degrees of freedom (see Fig. 3). One of the masses is assembled out of two masses, where the kinematic coupling constraint is situated. The states q_R and q_F^{co} with masses m_R and m_F^{co} coincide and it holds

$$q_1 = q_R = q_F^{co} \qquad \text{and} \qquad m_1 = m_R + m_F^{co}, \tag{55a}$$

as well as

$$q_2 = q_F^{in} \qquad \text{and} \qquad m_2 = m_F^{in}. \tag{55b}$$

The spring-damper elements are linear. The leftmost spring-damper element has stiffness parameter c_R and damping parameter d_R, whereas the middle and rightmost have stiffness and damping parameters c_F and d_F.

The equations of motion of the 2-mass spring-damper are given as follows

$$\dot{q}_1 = v_1, \tag{56a}$$

$$\dot{q}_2 = v_2, \tag{56b}$$

$$m_1\dot{v}_1 = -c_R q_1 - d_R v_1 + c_F(q_2 - q_1) + d_F(v_2 - v_1), \tag{56c}$$

$$m_2\dot{v}_2 = -c_F(q_2 - q_1) - d_F(v_2 - v_1) - c_F q_2 - d_F v_2, \tag{56d}$$

or written in matrix form

$$\underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix}}_{=\dot{x}} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-c_R - c_F}{m_1} & \frac{c_F}{m_1} & \frac{-d_R - d_F}{m_1} & \frac{d_F}{m_1} \\ \frac{c_F}{m_2} & \frac{-2c_F}{m_2} & \frac{d_F}{m_2} & \frac{-2d_F}{m_2} \end{pmatrix}}_{=:A} \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{bmatrix}}_{=:x}. \tag{57}$$

With initial value $x_0 = x(t_0)$, the analytical solution is given as

$$x(t) = e^{A(t-t_0)}x_0 \tag{58}$$

and serves as reference solution.

5.2 The decomposed subsystems

The 2-mass spring-damper model is decomposed as follows. At $q_1 = q_R = q_F^{co}$, it is split in half leading to the subsystems:

- multibody system with states q_R and corresponding mass m_R,
- flexible structure with states q_F^{co}, q_F^{in} and corresponding masses m_F^{co} and m_F^{in}.

The multibody system in this test model consists of one mass m_R with corresponding state q_R and a spring-damper element with stiffness and damping parameters c_R and d_R resulting in the linear ordinary differential equation

$$m_R \dot{v}_R = -c_R q_R - d_R v_R. \tag{59}$$

The flexible structure in this test model consists of two masses m_F^{co}, m_F^{in} with corresponding states q_F^{co}, q_F^{in} and two spring-damper elements with stiffness and damping parameters c_F and d_F.

The resulting linear equations of motion are given as ordinary differential equations

$$m_F^{co} \dot{v}_F^{co} = c_F(q_F^{in} - q_F^{co}) + d_F(v_F^{in} - v_F^{co}), \tag{60a}$$

$$m_F^{in} \dot{v}_F^{in} = -c_F(q_F^{in} - q_F^{co}) - d_F(v_F^{in} - v_F^{co}) - c_F q_F^{in} - d_F v_F^{in}. \tag{60b}$$

5.3 Coupling of the subsystems

As described in Sect. 2, we couple the subsystems by the kinematic coupling constraint

$$0 = q_R - q_F^{co} =: g_{co}(q_R, q_F^{co}). \tag{61}$$

Thus, the coupling states are q_R and q_F^{co}, the constraint gradient is

$$\frac{\partial g_{co}}{\partial (q_R, q_F)} = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \tag{62}$$

and the derivatives of g_{co} w.r.t. time result in the hidden constraints

$$0 = \frac{\partial g_{co}}{\partial (q_R, q_F)} \begin{bmatrix} v_R \\ v_F^{co} \\ v_F^{in} \end{bmatrix} \quad \text{and} \quad 0 = \frac{\partial g_{co}}{\partial (q_R, q_F)} \begin{bmatrix} \dot{v}_R \\ \dot{v}_F^{co} \\ \dot{v}_F^{in} \end{bmatrix}. \tag{63}$$

From (61) and (63), the prescribed coupling states are given as $q_F^{co} = q_R$, $v_F^{co} = v_R$, and $\dot{v}_F^{co} = \dot{v}_R$, i.e. ξ_q, ξ_v, and $\xi_{\dot{v}}$ equal identity. Therefore, we can simplify the general formulation of $u_F = y_R$ and it is sufficient to proceed with $u_F = y_R = [q_R, v_R, \dot{v}_R]^T = [q_F^{co}, v_F^{co}, \dot{v}_F^{co}]^T$. Moreover, we have $y_F = \Lambda$, skip (60a) as desribed in Sect. 2 and the coupled subsystems can be formulated as

$$\begin{bmatrix} \dot{q}_R \\ \dot{v}_R \end{bmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{c_R}{m_R} & -\frac{d_R}{m_R} \end{pmatrix}}_{=:A_R} \begin{bmatrix} q_R \\ v_R \end{bmatrix} + \underbrace{\begin{pmatrix} 0 \\ -\frac{1}{m_R} \end{pmatrix}}_{=:B_R} u_R, \tag{64}$$

$$\begin{bmatrix} \dot{q}_F^{in} \\ \dot{v}_F^{in} \end{bmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{2c_F}{m_F^{in}} & -\frac{2d_F}{m_F^{in}} \end{pmatrix}}_{=:A_F} \begin{bmatrix} q_F^{in} \\ v_F^{in} \end{bmatrix} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ \frac{c_F}{m_F^{in}} & \frac{d_F}{m_F^{in}} & 0 \end{pmatrix}}_{=:B_F} u_F. \tag{65}$$

The output equations for the multibody system are

$$y_R = \begin{bmatrix} q_R \\ v_R \\ \dot{v}_R \end{bmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{c_R}{m_R} & -\frac{d_R}{m_R} \end{pmatrix}}_{=:C_R} \begin{bmatrix} q_R \\ v_R \end{bmatrix} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ -\frac{1}{m_R} \end{pmatrix}}_{=:D_R} u_R \tag{66}$$

whereas for the flexible structure, depending on the evaluation of Λ, we get different expressions. If system (11) or (22) are used, which here is equivalent due to vanishing off-diagonal masses, it holds

$$y_F = \Lambda = \underbrace{\begin{pmatrix} -\frac{m_R c_F}{m_F^{co} + m_R} & -\frac{m_R d_F}{m_F^{co} + m_R} \end{pmatrix}}_{=:C_F} \begin{bmatrix} q_F^{in} \\ v_F^{in} \end{bmatrix} + \underbrace{\begin{pmatrix} \frac{-m_F^{co} c_R + m_R c_F}{m_F^{co} + m_R} & \frac{-m_F^{co} d_R + m_R d_F}{m_F^{co} + m_R} & 0 \end{pmatrix}}_{=:D_F} u_F. \tag{67}$$

Consequently, this approach leads to a zero-stable co-simulation, because $\frac{\partial y_F}{\partial \dot{v}_R} = 0$ is fulfilled and (36) shows that all eigenvalues are zero. An evaluation of (23) results in

$$y_F = \Lambda = \underbrace{\begin{pmatrix} -c_F & -d_F \end{pmatrix}}_{=:\tilde{C}_F} \begin{bmatrix} q_F^{in} \\ v_F^{in} \end{bmatrix} + \underbrace{\begin{pmatrix} c_F & d_F & m_F^{co} \end{pmatrix}}_{=:\tilde{D}_F} u_F \tag{68}$$

and zero-stability can, in general, not be guaranteed anymore.

5.4 Stability analysis of the test-model

As presented in Sect. 4, we want to analyze the stability of the 2-mass spring-damper co-simulation. With stiffness and damping parameters $c_R = 100$ N/m, $d_R = 1$ Ns/m, $c_F = 50$ N/m, and $d_F = 0.1$ Ns/m, we assemble the matrix (51) for the test example and check the spectral radius $\rho(\mathcal{S}(h))$ for varying masses.

In Fig. 4, the stability regions for exact force computation with (11) or (22) are shown for constant, linear, and quadratic extrapolation. Here, the coupled system is free of algebraic loops. The filled area denotes the region for unstable co-simulation, while in the blank area stable co-simulation is expected. One can observe, that the co-simulation becomes stable for sufficiently small macro steps, for all mass ratios and for all extrapolation methods.

In Fig. 5, for approximated forces from (23) and, thus, upcoming algebraic loop, the co-simulation is unstable for certain mass ratios, independent of the macro step size h. For an increasing order of extrapolation, the region of stable co-simulation shrinks.

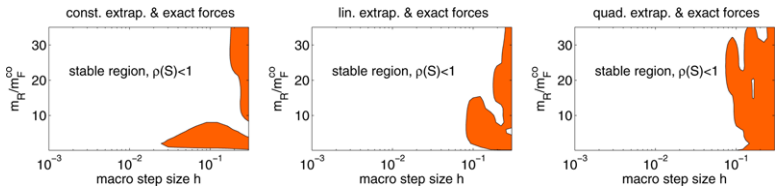

Fig. 4 Stable regions for exact forces depending on the mass ratio and the macro step size h with constant, linear, and quadratic extrapolation.

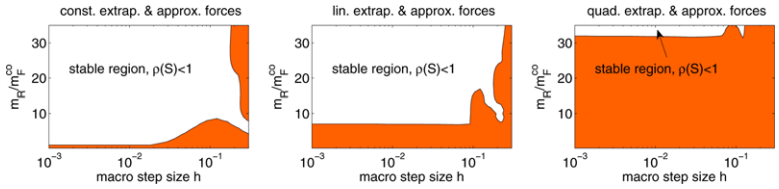

Fig. 5 Stable regions for approximated forces depending on the mass ratio and the macro step size h with constant, linear, and quadratic extrapolation.

In the case of approximated constraint forces, we have a closer look on the second term in the lower right block of $\mathcal{S}(h)$, as shown in Remark 4.1. We obtain the characteristic polynomial

$$0 = \mu^{4r}\mu^3 \left(\mu + \frac{1}{\mu} \left(\sum_{i=0}^{r} \frac{\alpha_i}{\mu^i} \right)^2 \frac{m_F^{co}}{m_R} \right). \tag{69}$$

Obviously, the ratio of the subsystem masses is decisive for the spectral radius. The spectral radii shown in Fig. 6 verify the thresholds in Fig. 5 of the mass ratio for stable co-simulations. Thus, even for very small macro step sizes h, we can not avoid instabilities which come from the loop in the input-output relation of the subsystems.

Moreover, we compare stability regions for varying stiffness and damping parameters of the multibody system and the flexible structure: we only consider linear extrapolation of inputs, the forces are computed approximately from (23) and we vary the parameters of the multibody system and the flexible structure separately (Fig. 7 and Fig. 8).

In Fig. 7, we see that changing the parameters of the multibody system does not influence the stable regions significantly. A higher multibody system stiffness slightly decreases the critical macro step size for stable simulations. Changing the damping parameter of the multibody system has almost no influence on stability, even for vanishing damping d_R.

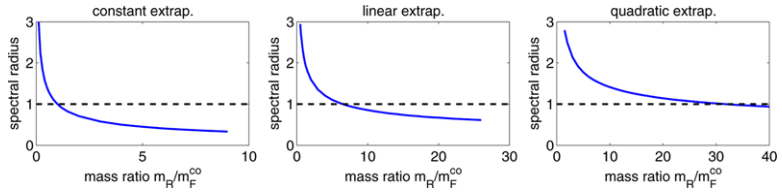

Fig. 6 Zero-stability (as in Remark 4.1) for constant, linear, and quadratic extrapolation depending on the mass ratio $\frac{m_R}{m_F^{co}}$.

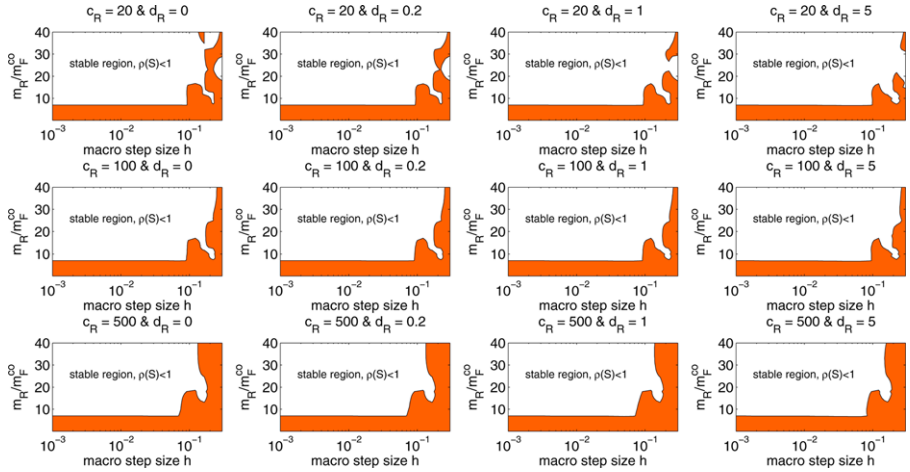

Fig. 7 Stable regions for varying stiffness and damping of the multibody system and linear extrapolation of inputs.

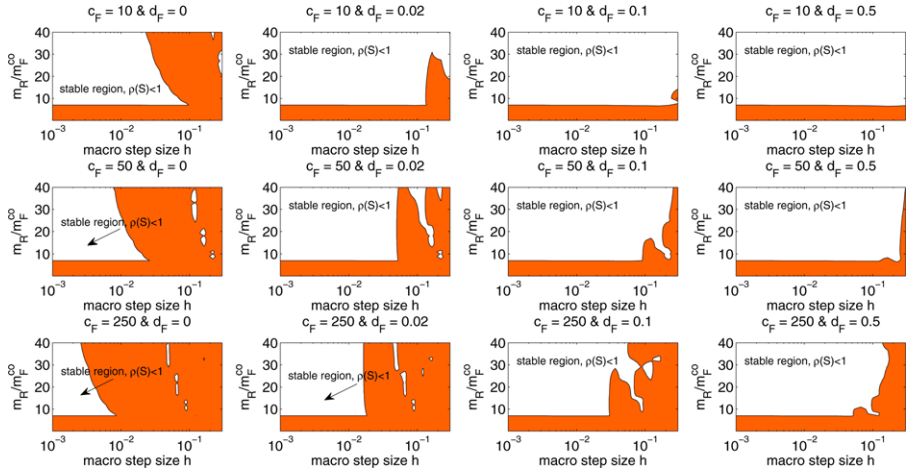

Fig. 8 Stable regions for varying stiffness and damping of the flexible structure and linear extrapolation of inputs.

On the other hand, in Fig. 8, both varying stiffness and damping parameter do result in different stable regions. For increasing stiffness, the critical macro step size decreases, while higher damping parameters increase the macro step size for which the simulation is stable. For vanishing damping d_F, very small h is required.

However, independent of stiffness and damping parameters, the mass ratio threshold for stable simulations remains unchanged and can not be improved with smaller macro step sizes. Thus, the subsystem's mass ratio has a dominating influence on the stability of the co-simulation algorithm.

Once the mass ratio allows stable co-simulation (or exact force evaluation is used), the combination of stiffness parameter c_F and damping parameter d_F of the flexible structure and the macro time step size h determines the stability of the co-simulation, as already described above. This is emphasized in Fig. 9.

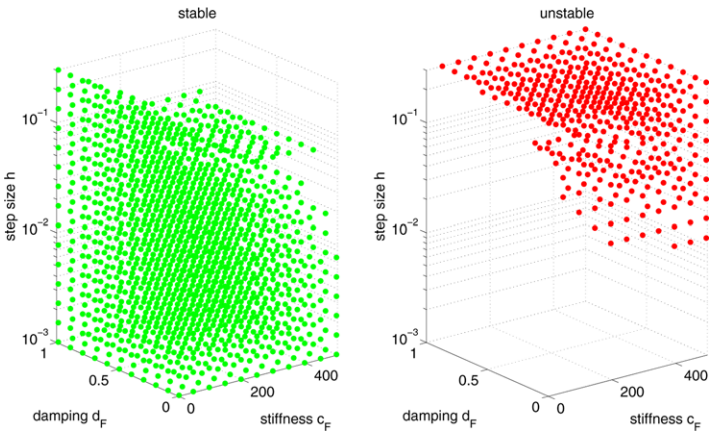

Fig. 9 3D-plots depending on parameters c_F, d_F, and macro step size h, for linear extrapolation of inputs and mass ratio $\frac{m_R}{m_F^{co}} = 10$: The left plot shows stable parameter configurations and the right plot shows unstable parameter configurations. Increasing stiffness leads to instabilities, while increasing damping stabilizes the simulation. Moreover, sufficiently small h results in stable simulation.

5.5 Numerical investigation of convergence

In order to analyze the convergence of the co-simulation method, some numerical experiments are carried out. To that end, the test model is implemented in MATLAB with system parameters as shown in Table 1.

The time integration of the subsystems are performed with the MATLAB integrator `ode45`. Thus, the convergence order of the co-simulation is not affected by the high-order time integration.

For the parallel co-simulation (Jacobi type), as explained in Sect. 3, the inputs of both subsystems need to be extrapolated. In the macro step from t_k up to t_{k+1}, the extrapolation with polynomials of order r, resp. s, uses $r+1$, resp. $s+1$, points from the past

$$u_{R,k-i}, \quad i = 0, ..., r, \quad \text{resp.} \quad u_{F,k-i}, \quad i = 0, ..., s, \tag{70}$$

to construct the polynomials $u_{R,k}^{\text{extrap},r}(t)$, resp. $u_{F,k}^{\text{extrap},s}(t)$. First, to compute the constraint forces $u_R = \Lambda$, we solve the system (11) or (22) . Later on, we also test the other option, using approximated forces from (23).

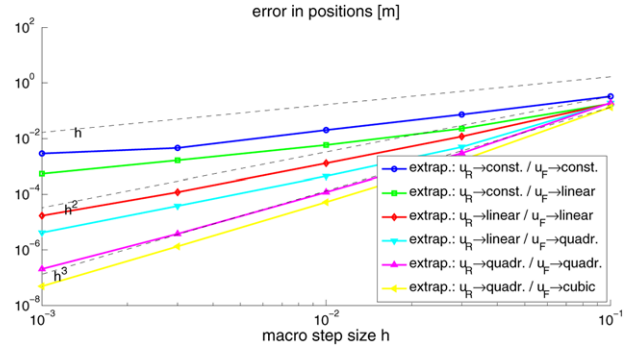

Fig. 10 Order of convergence depending on the extrapolation for exact constraint forces.

Table 1 Masses, stiffness, and damping parameters of the 2-mass spring-damper test model.

Parameter	Value
m_R	5 kg
m_F^{co}	0.5 kg
m_F^{in}	0.5 kg
c_R	100 N/m
d_R	1N s/m
c_F	50 N/m
d_F	0.1 Ns/m

The error plots in Fig. 10 show that extrapolation of the coupling terms is essential for the order of convergence. Especially the extrapolation of prescribed states $u_F = y_R = [q_R^T, v_R^T, \dot{v}_R^T]^T$ is important for fast convergence.

Next, we want to test the approximated constraint forces, where Λ is computed by (23). With masses from Table 1, the mass ratio becomes $\frac{m_R}{m_F^{co}} = 10$ and, looking at Fig. 6, the co-simulation with quadratic extrapolation should show instabilities. Indeed, the simulation becomes unstable in that case after 40 to 50 steps, as shown in Fig. 11.

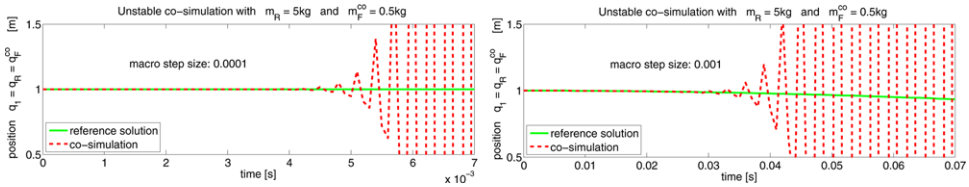

Fig. 11 Unstable co-simulation of the 2-mass spring-damper with quadratic extrapolation of inputs.

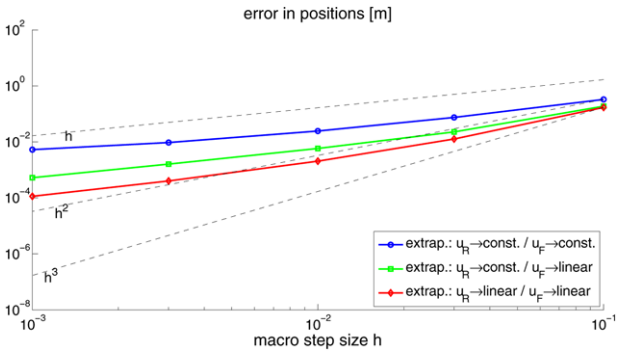

Fig. 12 Order of convergence for approximated constraint forces from Eq. (23).

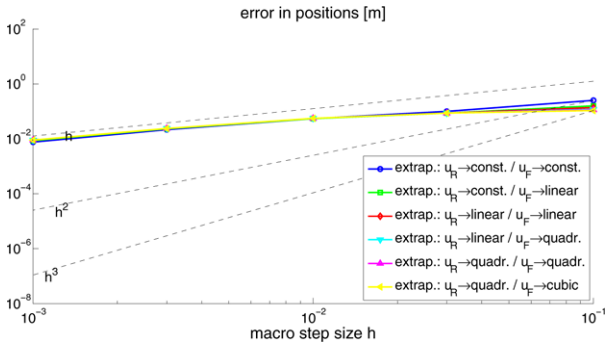

Fig. 13 Order of convergence for exact constraint forces, but with linear-implicit integrator for the flexible structure.

However, for constant and linear extrapolation the co-simulation is also stable with approximated constraint forces Λ from (23). To start the simulation, an initial constraint force Λ_0 is computed by using system (11) or (22) once in the preprocessing. The convergence behaves as for exact force computation, cf. Fig. 12.

To conclude this section, we want to give a remark on the time integrators. Commercial multibody system tools use accurate high-order integration methods. This is possible due to the efficient formulation of multibody systems. Usually, the flexible structure is more complex and fast simulation (up to realtime) is only possible with low-order methods. Very attractive for realtime applications are linear-implicit methods. They were applied successfully, e.g., in [2, 9].

If we integrate the flexible structure in our test model with a linear-implicit method, which does only one step per macro step, then the order of convergence is dominated by the low-order time integration (see Fig. 13).

That means, for realtime applications, where linear-implicit methods are used, it is not necessary to extrapolate with high-order schemes. Linear extrapolation for the prescribed coupling states u_F and even constant extrapolation for the constraint forces u_R are sufficient.

6 Application: Embedded cable model in a multibody system

The aim of this section is to apply the method to a more complex example. A 3D geometrically exact Cosserat rod model is embedded in a small multibody system. One end of the cable is coupled to the body with mass m_3. The other end is clamped at the wall.

The multibody system is simulated using local joint coordinates η_1, η_2 and η_3, as depicted in Fig. 14. The cable is modelled as one-dimensional structure with a linear viscoelastic material behaviour and allows all deformations: extension, bending, torsion and shearing. In particular, we consider a cable made of rubber with corresponding length $L = 1$ m, radius $r = 0.005$ m, density $\varrho = 1100$ kg/m^3, Young's modulus $E = 5 \cdot 10^6$ N/m^2, and Possion's ratio $\nu = 0.5$. For more details we refer to [14, 15].

Both subsystems are implemented in MATLAB and co-simulated in a parallel fashion, as explained in Sect. 3. One cable end and the body with mass m_3 are fully clamped, i.e. all degrees of freedom are constrained, such that g_{co} results in three translatory constrains of the form

$$0 = q_{1,R}^{co} - q_{1,F}^{co}\,, \qquad 0 = q_{2,R}^{co} - q_{2,F}^{co}\,, \qquad 0 = q_{3,R}^{co} - q_{3,F}^{co} \tag{71}$$

and three rotatory constrains formulated by scalar products of directors

$$0 = {d_{2,R}^{co}}^T \cdot d_{3,F}^{co}\,, \qquad 0 = {d_{3,R}^{co}}^T \cdot d_{1,F}^{co}, \quad \text{and} \quad 0 = {d_{1,R}^{co}}^T \cdot d_{2,F}^{co}. \tag{72}$$

Thus, the end-point positions and orientations of the cable are prescribed by the multibody system. The constraint forces acting on the multibody system are computed approximately according to (23).

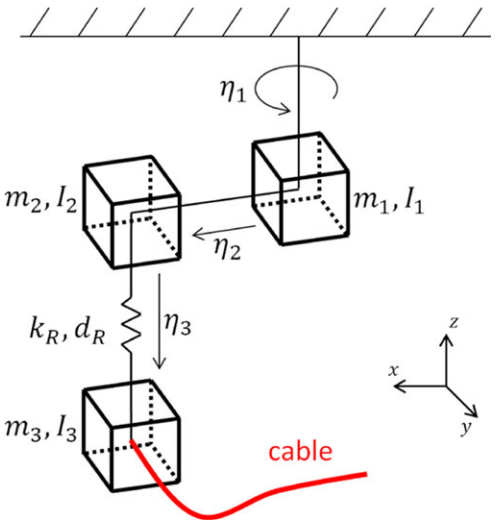

Fig. 14 Sketch of the multibody system.

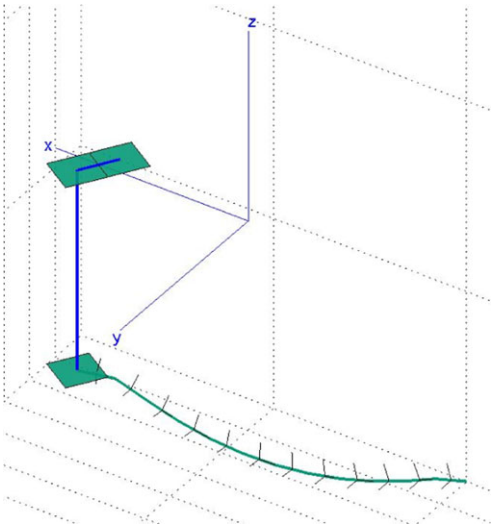

Fig. 15 MATLAB snapshot of the embedded cable.

The multibody system uses the MATLAB solver `ode15s` and the flexible structure is integrated with a linear-implicit method of order 1, which does one step per macro step. Consequently, the convergence of the co-simulation is at most of order 1 (see Fig. 16). Thus, it is sufficient to use linear extrapolation of the prescribed states and constant extrapolation for the constraint forces.

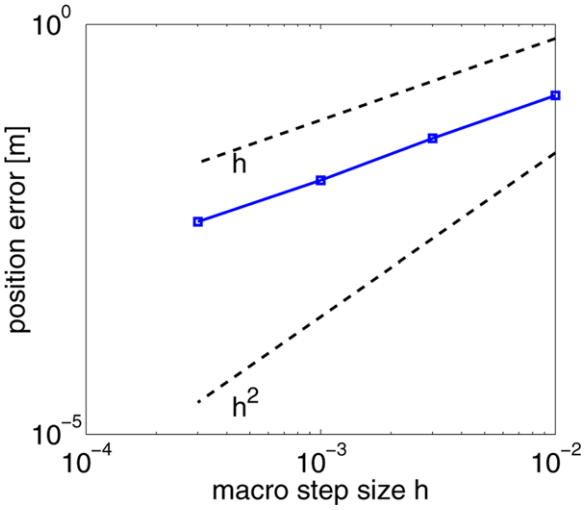

Fig. 16 Convergence of the co-simulation of the cable model and the multibody system: As reference solution, a monolithic simulation of the coupled subsystems was performed and the errors in the positions of the multibody system are plotted.

Moreover, we use the investigations from Sect. 4 about stability of our co-simulation approach. In order to apply the results from Sect. 4, we linearize the subsystems in an equilibrium point of the co-simulated system and, thus, receive the state space matrices.

Figure 17 shows the spectral radius of the term from Remark 4.1, i.e. the consideration of zero-stability including the extrapolation method, depending on the total mass ratio of the multibody system and the flexible cable model. As for the previous example, decreasing mass ratios, i.e. increasing cable mass (relative to the multibody system) lead to unstable co-simulations. Also, higher order extrapolation methods may result in an unstable system behaviour, as we have seen before.

For finite time step sizes, Fig. 18 shows the stable regions for constant, linear and quadratic extrapolation. Again, an increasing order of extrapolation shrinks the stable region concerning the mass ratio of the subsystems (also for constant extrapolation there is a very small mass ratio threshold which is hard to see here).

For all extrapolation cases and mass ratio above the critical threshold, we end up in the stable region for sufficiently small macro step sizes. But one should clarify, that since we consider the linearized systems here, the thresholds are only

valid for simulation with the linearized systems. This gives an idea on how to choose h for stable simulations with the original nonlinear system. In this example, the step size can be chosen larger than Fig. 18 suggests.

However, the mass ratio thresholds are very accurate, also for simulation with the nonlinear systems. In the presented co-simulation, the mass ratio, i.e. total mass of the multibody system divided by total mass of the cable, was approximately 7. If we decrease the mass ratio and apply quadratic extrapolation, the co-simulation becomes unstable for a mass ratio around 4. This perfectly confirms the stability investigations for quadratic extrapolation in Figs. 17 and 18. The same holds for linear and constant extrapolation, if we successively decrease the mass ratio.

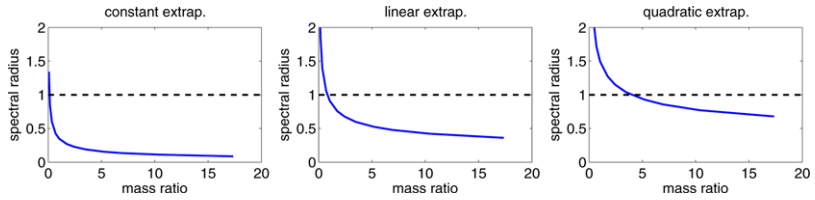

Fig. 17 Zero-stability for constant, linear, and quadratic extrapolation depending on the total mass ratio of MBS and cable "$\frac{m_R}{m_F}$".

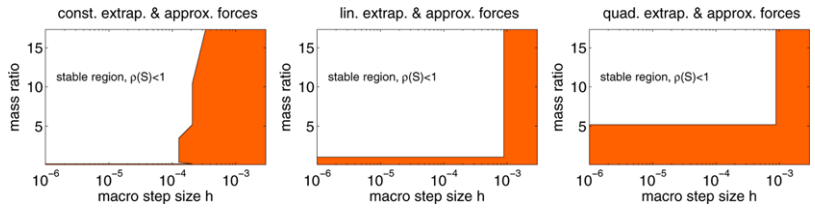

Fig. 18 Stable regions for finite time steps (with const., lin., and quadr. extrapolation) depending on step size h and mass ratio "$\frac{m_R}{m_F}$".

7 Conclusion

In this work, we presented an efficient force-displacement co-simulation approach that uses an algebraic constraint for the kinematic coupling, and we have shown its analytical equivalence to the monolithic system simulation. This approach is advantageous over a bushing-element, since no artificial stiffness is introduced and it is easy to formulate more complex coupling joints.

Although constraint forces are used, the fact that we end up with a force-displacement co-simulation avoids drift-off in the kinematic constraint. The mentioned constraint forces can be derived exactly or approximately in a much faster computation. The force approximation leads to some instability issues. However, they do not cause problems if the subsystems are assembled in the right order, concerning the mass ratio.

After the derivation of the coupling equations, we tested our approach on two examples. First, a 2-mass spring-damper model was modified such that we could apply our force-displacement co-simulation. A number of numerical experiments have been performed to analyze the stability and convergence of our approach. Later on, we faced a more complex problem and used our method to embed a cable model in a multibody system, showing the potential of our coupling method.

It is worth pointing out, that the application of the presented concept is not restricted to the simulation of flexible multibody systems, we have mainly been considering here, but it can be used for various problems where subsystems are coupled kinematically.

In future work, several aspects would be interesting. From a theoretical point of view, the incorporation of subsystems, which are formulated as differential algebraic equations instead of ordinary differential equations, is an interesting issue. Also, the influence of the discretization of the flexible structure on the stability of the coupling approach is worth to examine. From a practical viewpoint, using the presented approach to embed further flexible structures, e.g. shells, in multibody systems should be considered.

References

[1] FMI: Functional Mock-up Interface for Model Exchange and Tool Coupling Version 2.0, https://www.fmi-standard.org/downloads.
[2] M. Arnold, B. Burgermeister, and A. Eichberger, Linearly implicit time integration methods in real-time applications: DAEs and stiff ODEs, Multibody Syst. Dyn. **17**(2-3), 99–117 (2007).

[3] M. Arnold and M. Günther, Preconditioned dynamic iteration for coupled differential-algebraic systems, BIT Numer. Math. **41**(1), 001–025 (2001).
[4] A. Bartel, M. Brunk, M. Günther, and S. Schöps, Dynamic iteration for coupled problems of electric circuits and distributed devices, SIAM J. Sci. Comput. **35**(2), B315–B335 (2013).
[5] M. Busch, Zur effizienten Kopplung von Simulationsprogrammen, PhD thesis, Universität Kassel (2012).
[6] M. Busch and B. Schweizer, Numerical stability and accuracy of different co-simulation techniques: Analytical investigations based on a 2-dof test model, in: The 1st Joint International Conference on Multibody System Dynamics, May 25–27, 2010, Lappeenranta, Finland (2010).
[7] M. Busch and B. Schweizer, Coupled simulation of multibody and finite element systems: an efficient and robust semi-implicit coupling approach, Arch. Appl. Mech. **82**(6), 723–741 (2012).
[8] R. R. Craig. and M. C. C. Bampton Jr, Coupling of substructures for dynamic analyses, AIAA J. **6**(7), 1313–1319 (1968).
[9] L. Engelhardt, M. Burger, and G. Bitsch, Real-time simulation of multibody-systems for on-board applications, Tech. Rep. 193, Fraunhofer ITWM (2010).
[10] B. Esterl, T. Butz, B. Simeon, and B. Burgermeister, Real-time capable vehicle–trailer coupling by algorithms for differential-algebraic equations, Veh. Syst. Dyn. **45**(9), 819–834 (2007).
[11] C. A. Felippa, K. Park, and C. Farhat, Partitioned analysis of coupled mechanical systems, Comput. Methods Appl. Mech. Eng. **190**(24), 3247–3270 (2001).
[12] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer Series in Computational Mathematics Vol. 14 (Springer-Verlag, Berlin Heidelberg, 1996).
[13] R. Kübler and W. Schiehlen, Two methods of simulator coupling, Math. Comput. Model. Dyn. Syst. **6**(2), 93–113 (2000).
[14] H. Lang, J. Linn, and M. Arnold, Multibody dynamics simulation of geometrically exact Cosserat rods, Multibody Syst. Dyn. **25**(3), 285–312 (2011).
[15] J. Linn and T. Stephan, Fast simulation of quasistatic cable deformations using discrete rod models, Tech. Rep. 144, Fraunhofer ITWM, 2008.
[16] W. Schiehlen, A. Rükgauer, and T. Schirle, Force coupling versus differential algebraic description of constrained multibody systems, Multibody Syst. Dyn. **4**(4), 317–340 (2000).
[17] T. Schierz and M. Arnold, Stabilized overlapping modular time integration of coupled differential-algebraic equations, Appl. Numer. Math. **62**(10), 1491–1502 (2012).
[18] B. Schweizer and D. Lu, Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints, Syst. Dyn. **34**(2), 129–161 (2015).
[19] B. Simeon, Computational Flexible Multibody Dynamics (Springer-Verlag, Berlin Heidelberg, 2013).
[20] R. Wehage and E. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, J. Mech. Des. **104**(1), 247–255 (1982).

