

Implicit co-simulation method for constraint coupling with improved stability behavior

T. Meyer¹ · P. Li¹ · D. Lu¹ · B. Schweizer¹

Received: 13 April 2017 / Accepted: 26 April 2018
© Springer Science+Business Media B.V., part of Springer Nature 2018

Abstract This paper deals with a novel co-simulation approach for coupling mechanical subsystems in time domain. The submodels are assumed to be coupled by algebraic constraint equations. In contrast to well-known coupling techniques from the literature, the here presented index-1 approach uses a special technique for approximating the coupling variables so that the constraint equations together with the hidden constraints on velocity and acceleration level can be enforced simultaneously at the communication time points. The method discussed here uses second- and third-order approximation polynomials. Because of the high approximation order, the numerical errors are very small, and a good convergence behavior is achieved. A stability analysis is carried out, and it is shown that—despite the fact that higher-order approximation polynomials are applied—also a good numerical stability behavior is observed. Different numerical examples are presented, which illustrate the practical application of the approach.

Keywords Solver coupling · Co-simulation · Algebraic constraints · Parallelization · Mechanical systems · Multibody systems

1 Introduction

There is a continuously increasing number of problems, where solver coupling or co-simulation techniques are applied successfully [11]. Co-simulation approaches are used to parallelize dynamical systems to speed up the simulation time; see, e.g., [18, 19, 22, 24, 27, 28, 37, 42]. Therefore, a global model is split into different submodels. The submodels are integrated on different CPUs and coupled by an appropriate co-simulation approach. Solver coupling approaches are also useful for simulating multiphysical systems [8, 14, 30, 39]. There exist different fields of applications, where co-simulation has been advantageously applied to solve multidisciplinary problems. Finite-element models have been coupled with

✉ T. Meyer
meyer@ad.tu-darmstadt.de

¹ Department of Mechanical Engineering, Institute of Applied Dynamics, Technical University Darmstadt, Otto-Bernd-Straße 2, 64287 Darmstadt, Germany

multibody models to analyze catenary/pantograph-contact problems [2, 3, 29] for investigating interaction problems of vehicles with flexible structures. Coupling of boundary-element with multibody models is discussed in [45] in the framework of contact simulation. Solver coupling is also frequently used for simulating fluid/structure interaction problems; see, e.g., [1]. Coupled mechanical/hydraulic systems are examined in [25]. Application of co-simulation techniques in the field of vehicle dynamics is treated in [7, 21, 26]. Coupled electromechanical systems are analyzed in [44]. The coupled simulation of particle models with multibody systems is investigated in [9, 20, 38].

The coupling between two subsystems can either be accomplished by algebraic constraint equations or by constitutive equations. In the former case, *Lagrange* multipliers are used to couple the subsystems [5, 15, 17, 32, 34, 40, 41, 43], whereas, in the latter case, applied forces and torques are applied to realize the connection between the subsystems [1, 3, 6, 12, 13, 33, 35]. Solver coupling based on algebraic constraints is usually carried out with implicit or linear-implicit co-simulation techniques because of their improved stability behavior compared to explicit coupling methods. However, also explicit coupling schemes have been presented in literature for constraint coupling problems [15, 31]. Using physical force/torque laws, i.e., constitutive equations, to couple the subsystems, explicit techniques are frequently applied; for stiff problems, however, implicit schemes might be more efficient and stable [35].

In the framework of a co-simulation approach, the global model is decomposed into two or several submodels. The subsystems are connected by coupling variables. Additionally, a communication time grid is defined by introducing the communication time points $T_0, \dots, T_N, \dots, T_{end}$. To keep the representation concise, in this work, the communication-step size $H = T_{N+1} - T_N$ is assumed to be constant. Coupling approaches with nonequidistant communication time grids are discussed, for instance, in [4]. The coupling variables are only exchanged at the communication time points. Between two communication time points, the subsystem solvers integrate independently. To carry out the subsystem integration, the coupling variables have to be approximated by appropriate extrapolation/interpolation functions [10].

Here we consider a solver coupling technique using algebraic constraint equations. Different methods have been proposed in the literature to handle co-simulation problems where algebraic constraints are used to couple the submodels. A direct index-3 approach is discussed in [41, 43]. Numerical studies have, however, shown that the direct index-3 implementation may produce artificial oscillations and chattering phenomena; see, e.g., [40]. An explicit co-simulation method incorporating the well-established *Baumgarte* stabilization technique is suggested in [15]. This explicit approach may be implemented very efficiently; due to its explicit character, the method suffers, however, from a reduced numerical stability. *Baumgarte* stabilization in connection with an implicit implementation is analyzed in [32, 36]. The implicit approach shows, on the one hand, a significantly improved stability behavior. On the other hand, the implicit implementation requires a repetition of the macro step so that the subsystem solvers have to be reinitialized at the previous macro time point. Since solver reinitialization may not be directly supported by every commercial software code, application of implicit coupling schemes that require solver reinitialization may be limited. Solver coupling on the basis of a classical projection approach is treated in [36], where also a detailed stability analysis is presented. Compared with the *Baumgarte* implementation, solver coupling based on a projection technique has the advantage that no artificial parameters have to be defined. Solver coupling with constraints in connection with the parallelization of multibody systems is treated in [24]; this method has similarities with the *Baumgarte* stabilization, since the approach also introduces artificial dynamics. The well-known *Gear–Gupta–Leimkuhler* stabilization technique for DAE systems can also be used

in connection with solver coupling; see [34]. A stability analysis of such an index-2 co-simulation approach shows, however, that numerical stability can only be guaranteed for the case where the approximation polynomials are constant. Using higher-order approximation polynomials requires additional stabilization techniques. A sequential coupling approach on index-1 level is presented and analyzed in [5, 17]; a stability analysis exhibits that the method produces stable results if a special contractivity condition is fulfilled. An interesting explicit co-simulation approach tailored for constraint coupling problems has recently been presented in [31]. This approach allows a very straightforward implementation. Because of the explicit nature of this method, the numerical stability is reduced compared with implicit schemes. Applying a proper subsystem decomposition, the methods produce stable results.

Regarding co-simulation techniques, two main features are of special interest, namely the numerical stability and the convergence behavior of the coupling technique. In connection with constraint coupling approaches, most frequently, constant approximation polynomials are used.

Applying higher-order approximation functions for the coupling variables increases the accuracy of the method but usually entails a significantly reduced numerical stability, so that the method may practically not be used. The solver coupling approach presented here is based on second- and third-order approximation polynomials but still shows a good numerical stability behavior. Especially, the third-order approach is very stable and even outperforms the numerical stability of many of the mentioned lower-order methods. The coupling technique analyzed here is an index-1 approach without numerical drift-off effect, since the coupling conditions on position, velocity, and acceleration level are simultaneously taken into account at the communication time points. In contrast, the co-simulation approaches examined in [15, 24, 32, 36] introduce additional dynamics and require special parameters to be chosen properly. Co-simulation techniques incorporating the *Gear–Gupta–Leimkuhler* stabilization are index-2 methods, so that the coupling constraints are not fulfilled on acceleration level.

The second main advantage of the presented method is its high convergence order. The numerical examples show that the local error convergence is of order $\mathcal{O}(H^6)$ for the case where quadratic approximation polynomials are used and of order $\mathcal{O}(H^7)$ for cubic polynomials.

Although a good numerical stability and a high convergence order are achieved, the presented method also has different drawbacks. The main disadvantage of the presented approach compared to the methods discussed lies in the fact that the implementation of the method is more complicated. Since the method is linear implicit and based on a predictor/corrector technique, a coupling *Jacobian* matrix is necessary, which requires the partial derivatives of the state variables of the coupling bodies with respect to the coupling variables. This *Jacobian* is very small for the case where only two subsystems are coupled, so that the partial derivatives can be calculated very efficiently in parallel with the predictor step. If multiple subsystems are coupled simultaneously, then the size of the *Jacobian* increases, so that a parallel calculation of the coupling *Jacobian* may not be carried out in parallel with the predictor step. Then the efficiency of the method is reduced significantly. A second drawback of the method is that the time-step size for the subsystem integration has to be chosen properly to correctly calculate the coupling *Jacobian*.

The manuscript is structured as follows: In Sect. 2, we introduce the higher-order index-1 coupling approach with the help of a co-simulation test model. The co-simulation approach is generalized in Sect. 3 for the case where two arbitrary mechanical systems are coupled by algebraic constraint equations (i.e., by rigid joints). Numerical examples including convergence plots are presented in Sect. 4. The paper is summarized in Sect. 5. A detailed stability analysis for the co-simulation approach is presented in Appendix A.

2 Index-1 coupling approach: co-simulation test model

2.1 Co-simulation test model

A general linear co-simulation test model for solver coupling approaches based on algebraic constraint equations is the 1-degree-of-freedom oscillator shown in Fig. 1. A rigid massless link—mathematically described by the algebraic constraint $x_2 - x_1 = 0$ —connects two single-mass oscillators (masses m_1, m_2 ; spring constants c_1, c_2 ; damping coefficients d_1, d_2).

The coordinates x_1, x_2 describe the displacements of the two masses, and v_1, v_2 specify the corresponding velocities. The springs are stress-free for $x_1 = x_2 = 0$. The overall system is represented by the following autonomous index-3 DAE system

$$\begin{aligned} \dot{x}_1 &= v_1, \\ \dot{v}_1 &= -\frac{c_1}{m_1}x_1 - \frac{d_1}{m_1}v_1 + \frac{\lambda_c}{m_1}, \\ \dot{x}_2 &= v_2, \\ \dot{v}_2 &= -\frac{c_2}{m_2}x_2 - \frac{d_2}{m_2}v_2 - \frac{\lambda_c}{m_2}, \\ g_{c\lambda} &:= x_2 - x_1 = 0, \end{aligned} \tag{1}$$

where λ_c represents the *Lagrange* multiplier. The initial conditions for the state variables are given by

$$\begin{aligned} x_1(t=0) &= x_{1,0} = x_0, & v_1(t=0) &= v_{1,0} = v_0, \\ x_2(t=0) &= x_{2,0} = x_0, & v_2(t=0) &= v_{2,0} = v_0, \end{aligned} \tag{2}$$

and the initial condition for λ_c is determined by

$$\lambda_c(t=0) = \lambda_{c,0} = \frac{m_1 \cdot m_2}{m_1 + m_2} \cdot \left[\left(\frac{c_1}{m_1} - \frac{c_2}{m_2} \right) \cdot x_0 + \left(\frac{d_1}{m_1} - \frac{d_2}{m_2} \right) \cdot v_0 \right]. \tag{3}$$

2.2 Co-simulation procedure

Applying a force/force-decomposition approach (see, e.g., [36]), the overall system (Eq. (1)) is split into two subsystems so that both subsystems are force-driven single-mass oscillators; see Fig. 2. The single-mass oscillators are excited by the coupling force λ_c .

Fig. 1 Co-simulation test model for solver coupling with algebraic constraints

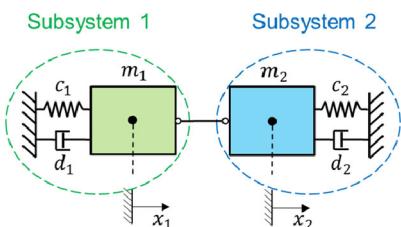
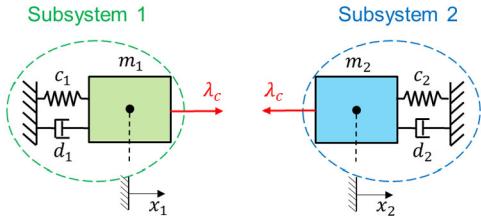


Fig. 2 Force/force-decomposition of the co-simulation test model



After the decomposition, the test model is described by the following overdetermined system:

Subsystem 1:

$$\begin{aligned}\dot{x}_1 &= v_1, \\ \dot{v}_1 &= -\frac{c_1}{m_1}x_1 - \frac{d_1}{m_1}v_1 + \frac{\lambda_c}{m_1}\end{aligned}\quad (\text{a}),$$

Subsystem 2:

$$\begin{aligned}\dot{x}_2 &= v_2, \\ \dot{v}_2 &= -\frac{c_2}{m_2}x_2 - \frac{d_2}{m_2}v_2 - \frac{\lambda_c}{m_2}\end{aligned}\quad (\text{b}),$$

Coupling conditions:

$$g_{c\lambda} := x_2 - x_1 = 0, \quad (\text{c1})$$

$$\dot{g}_{c\lambda} := v_2 - v_1 = 0, \quad (\text{c2})$$

$$\ddot{g}_{c\lambda} := \dot{v}_2 - \dot{v}_1 = 0 \quad (\text{c3}).$$

Note that the coupling conditions have been complemented by the hidden constraints $\dot{g}_{c\lambda}$ and $\ddot{g}_{c\lambda}$, which are required for the presented coupling technique. We regard the general macro time step from T_N to T_{N+1} to illustrate the coupling approach. At the communication time point T_N , the state variables and the coupling variable are given by

$$\begin{aligned}x_1(t = T_N) &= x_{1,N}, & v_1(t = T_N) &= v_{1,N}, & (\text{a}) \\ x_2(t = T_N) &= x_{2,N}, & v_2(t = T_N) &= v_{2,N}, & (\text{b}) \\ \lambda_c(t = T_N) &= \lambda_{c,N}\end{aligned}\quad (\text{5})$$

The coupling force $\lambda_c(t)$ is approximated either by a polynomial of degree $k = 2$ or by a polynomial of degree $k = 3$. Using quadratic approximation polynomials results in a discontinuous approximation at the communication time points; with cubic polynomials, the approximation is continuous. In both cases, we assume that the approximation polynomial contains three parameters, namely $\beta_{1,N+1}, \beta_{2,N+1}, \beta_{3,N+1}$, which are collected in the vector $\boldsymbol{\beta}_{N+1}$. These parameters are used to enforce the three coupling conditions, namely the coupling conditions on position, velocity, and acceleration level. For the general macro time step $T_N \rightarrow T_{N+1}$, the approximation polynomials are defined by

$$\begin{aligned}\lambda_c(t) &= \lambda_{c,N+1}(\boldsymbol{\beta}_{N+1}; t) \\ &= \begin{cases} \beta_{1,N+1} \cdot p_1(\tau) + \beta_{2,N+1} \cdot p_2(\tau) + \beta_{3,N+1} \cdot p_3(\tau) & (k = 2) \\ \lambda_{c,N} + \beta_{1,N+1} \cdot p_1(\tau) + \beta_{2,N+1} \cdot p_2(\tau) + \beta_{3,N+1} \cdot p_3(\tau) & (k = 3) \end{cases} \\ &\quad \left(\tau = \frac{t - T_N}{T_{N+1} - T_N} \right),\end{aligned}\quad (6)$$

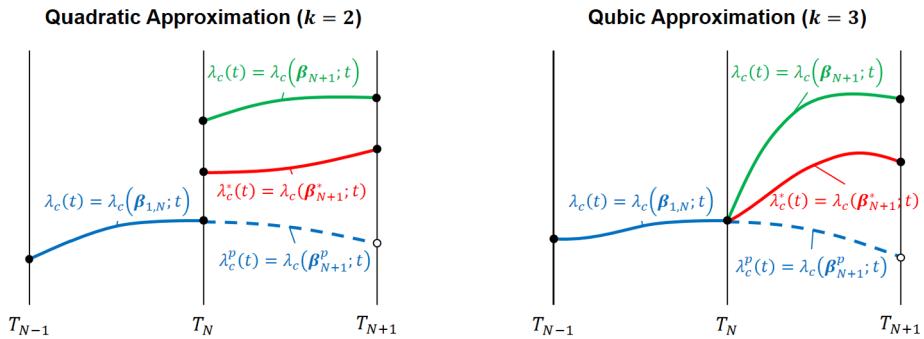


Fig. 3 Approximation polynomials ($k = 2$ and $k = 3$): predictor polynomial $\lambda_c^p(t)$, arbitrary polynomial $\lambda_c^*(t)$, and corrector polynomial $\lambda_c(t)$

where the three polynomials $p_j(\tau)$ ($j = 1, 2, 3$) are given by

$$\begin{aligned} p_j(\tau) &= \sum_{i=1}^3 \alpha_{i,j} \cdot \tau^{i+k-3} \\ &= \begin{cases} \alpha_{1,j} + \alpha_{2,j} \cdot \tau + \alpha_{3,j} \cdot \tau^2 & (k=2) \\ \alpha_{1,j} \cdot \tau + \alpha_{2,j} \cdot \tau^2 + \alpha_{3,j} \cdot \tau^3 & (k=3) \end{cases} \quad (j=1, 2, 3). \end{aligned} \quad (7)$$

The values $\alpha_{i,j} \in \mathbb{R}$ ($i, j = 1, 2, 3$) are user-defined coefficients, which have to be chosen in such a way that the three polynomials $p_j(\tau)$ are linearly independent.

Since the space of all polynomials of degree $k \leq 2$ is a three-dimensional vector space, each quadratic polynomial can be represented uniquely as a linear combination of the three polynomials of Eq. (7). The space of all polynomials $p(\tau)$ of degree $k \leq 3$ with $p(0) = 0$ is a three-dimensional subspace of the four-dimensional vector space of all polynomials of degree $k \leq 3$. Hence, all polynomials $\lambda_c(t)$ of degree $k \leq 3$ with $\lambda_c(T_N) = \lambda_{c,N}$ are represented uniquely by Eq. (6).

The co-simulation approach considered here is based on a three-stage predictor/corrector approach. For the three steps, three different approximation polynomials are used, namely the predictor polynomial $\lambda_c^p(t) = \lambda_{c,N+1}(\beta_{N+1}^p; t)$, the general polynomial $\lambda_c^*(t) = \lambda_{c,N+1}(\beta_{N+1}^*; t)$, and the corrector polynomial $\lambda_c(t) = \lambda_{c,N+1}(\beta_{N+1}; t)$. It should be mentioned that in the following representation, predicted variables are indicated with a superscript p (e.g., $x_{1,N+1}^p$). General variables are indicated with an asterisk (e.g., $x_{1,N+1}^*$). Variables without superscript are assumed to be corrected variables (e.g., $x_{1,N+1}$). The three different polynomials are illustrated in Fig. 3 for $k = 2$ and $k = 3$.

The presented coupling approach uses a predictor/corrector technique. To describe the method, we consider the general macro time step from T_N to T_{N+1} . We ride on the assumption that both subsystems are integrated independently between the macro time points using extrapolated/interpolated coupling variables. At the beginning of the macro time step, the state variables of the two subsystems are assumed to be known. We also assume that the corrected approximation polynomial of the previous macro time step is known. The three basic steps of the co-simulation approach are explained next.

Step 1: Predictor step

- With the corrected approximation polynomial $\lambda_c(t) = \lambda_{c,N}(\beta_N; t)$ of the previous communication time step $T_{N-1} \rightarrow T_N$ (β_N collects the corrected polynomial parameters

of the previous macro step), it is straightforward to generate the predictor polynomial

$$\lambda_c^p(t) = \begin{cases} \lambda_{c,N+1}(\boldsymbol{\beta}_N; t + H) & (k = 2) \\ \lambda_{c,N-1} - \lambda_{c,N} + \lambda_{c,N+1}(\boldsymbol{\beta}_N; t + H) & (k = 3) \end{cases} \quad (8)$$

for the time interval $[T_N, T_{N+1}]$ by a simple continuation; see Fig. 3. The predicted polynomial parameters $\boldsymbol{\beta}_{N+1}^p$ are determined uniquely. Note that $H = T_{N+1} - T_N$ represents the macro step size and $\lambda_{c,N} = \lambda_{c,N}(\boldsymbol{\beta}_N; T_N)$ the coupling force at the communication time point T_N .

- Using the predicted (extrapolated) coupling force

$$\lambda_c^p(t) = \lambda_{c,N+1}(\boldsymbol{\beta}_{N+1}^p; t), \quad (9)$$

an integration of subsystem 1 and subsystem 2 from T_N to T_{N+1} with the initial conditions (5)(a) yields the predicted state and coupling variables

$$x_{1,N+1}^p, v_{1,N+1}^p, x_{2,N+1}^p, v_{2,N+1}^p, \lambda_{c,N+1}^p = \lambda_c^p \quad (10)$$

at the communication time point T_{N+1} . Inserting the predicted state variables into the equations of motion (4)(a) and (b) yields the predicted accelerations at the communication-time point T_{N+1} , namely

$$\dot{v}_{1,N+1}^p, \dot{v}_{2,N+1}^p. \quad (11)$$

- With Eq. (4)(c), we get the subsequent residuals of the predictor step

$$\begin{aligned} g_{c\lambda,N+1}^p &:= x_{2,N+1}^p - x_{1,N+1}^p, \\ \dot{g}_{c\lambda,N+1}^p &:= v_{2,N+1}^p - v_{1,N+1}^p, \\ \ddot{g}_{c\lambda,N+1}^p &:= \dot{v}_{2,N+1}^p - \dot{v}_{1,N+1}^p. \end{aligned} \quad (12)$$

Step 2: Calculation of corrected polynomial parameters

- Integrating the subsystems with the predictor polynomial yields predicted state and acceleration variables at T_{N+1} , which usually do not fulfill the constraint and its derivatives. Within the linear implicit method considered here, the macro step is repeated with a corrected (i.e., optimized) polynomial so that the linearized constraint and its derivatives are enforced at T_{N+1} . Therefore, the constraint equation is linearized at T_{N+1} by a Taylor expansion using the predictor variables at T_{N+1} as expansion point. The crucial point is the calculation of the *Jacobian* matrix for the linearized coupling equations. To clearly define the calculation of this *Jacobian* matrix, we define a “general coupling force” and corresponding “general variables”. In a practical implementation of the considered co-simulation approach, only the *Jacobian* matrix has to be calculated, but not the “general variables”. Introduction of the “general variables” is only accomplished for the reason of a clear representation.

- Making use of the general coupling force

$$\lambda_c^*(t) = \lambda_{c,N+1}(\boldsymbol{\beta}_{N+1}^*; t), \quad (13)$$

an integration of both subsystems yields the general variables

$$x_{1,N+1}^*, v_{1,N+1}^*, x_{2,N+1}^*, v_{2,N+1}^*, \lambda_{c,N+1}^* = \lambda_c^* \quad (t = T_{N+1}) \quad (14)$$

at the communication time point T_{N+1} , which depend on β_{N+1}^* . Note that β_{N+1}^* terms arbitrary polynomial parameters. Inserting the general state variables from Eq. (14) into the equations of motion (4)(a) and (b) yields general accelerations at the communication-time point T_{N+1} , i.e.,

$$\dot{v}_{1,N+1}^*, \dot{v}_{2,N+1}^*. \quad (15)$$

- Regarding the fixed time point T_{N+1} , the residuals $g_{c\lambda,N+1}$, $\dot{g}_{c\lambda,N+1}$, and $\ddot{g}_{c\lambda,N+1}$ are regarded as functions of the polynomial parameters β_{N+1}^* , i.e.,

$$\begin{aligned} g_{c\lambda,N+1}(\beta_{N+1}^*) &:= x_{2,N+1}(\beta_{N+1}^*) - x_{1,N+1}(\beta_{N+1}^*) = x_{2,N+1}^* - x_{1,N+1}^*, \\ \dot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &:= v_{2,N+1}(\beta_{N+1}^*) - v_{1,N+1}(\beta_{N+1}^*) = v_{2,N+1}^* - v_{1,N+1}^*, \\ \ddot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &:= \dot{v}_{2,N+1}(\beta_{N+1}^*) - \dot{v}_{1,N+1}(\beta_{N+1}^*) = \dot{v}_{2,N+1}^* - \dot{v}_{1,N+1}^*. \end{aligned} \quad (16)$$

- Differentiating the residuals with respect to β_{N+1}^* yields the gradients

$$\frac{\partial g_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*}, \quad \frac{\partial \dot{g}_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*}, \quad \frac{\partial \ddot{g}_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*}, \quad (17)$$

which are constant since the state variables of Eq. (14) only depend linearly on β_{N+1}^* .

- Using the residuals of the predictor step and the partial derivatives of the constraints, we will calculate corrected polynomial parameters to enforce the coupling conditions (4)(c) at the communication time point T_{N+1} . Since the state variables $x_{1,N+1}^*$, $x_{2,N+1}^*$, $v_{1,N+1}^*$, $v_{2,N+1}^*$ and the accelerations $\dot{v}_{1,N+1}^*$, $\dot{v}_{2,N+1}^*$ depend only linearly on β_{N+1}^* , Eq. (16) can be rewritten as

$$\begin{aligned} g_{c\lambda,N+1}(\beta_{N+1}^*) &:= g_{c\lambda,N+1}^p + \left. \frac{\partial g_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*} \right|_{\beta_{N+1}^p} \cdot (\beta_{N+1}^* - \beta_{N+1}^p), \\ \dot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &:= \dot{g}_{c\lambda,N+1}^p + \left. \frac{\partial \dot{g}_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*} \right|_{\beta_{N+1}^p} \cdot (\beta_{N+1}^* - \beta_{N+1}^p), \\ \ddot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &:= \ddot{g}_{c\lambda,N+1}^p + \left. \frac{\partial \ddot{g}_{c\lambda,N+1}(\beta_{N+1}^*)}{\partial \beta_{N+1}^*} \right|_{\beta_{N+1}^p} \cdot (\beta_{N+1}^* - \beta_{N+1}^p). \end{aligned} \quad (18)$$

- By setting

$$\begin{aligned} g_{c\lambda,N+1}(\beta_{N+1}^*) &= 0, \\ \dot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &= 0, \\ \ddot{g}_{c\lambda,N+1}(\beta_{N+1}^*) &= 0, \end{aligned} \quad (19)$$

we get the corrected polynomial parameters β_{N+1} . It should be pointed out that different variables have been used for the general parameters β_{N+1}^* and the corrected parameters β_{N+1} , which represent the roots of Eq. (18).

Step 3: Corrector step

- Using the corrector (interpolation) polynomial

$$\lambda_c(t) = \lambda_{c,N+1}(\beta_{N+1}; t) \quad (20)$$

with the corrected parameters β_{N+1} , an integration of both subsystems yields the corrected state and coupling variables

$$x_{1,N+1}, v_{1,N+1}, x_{2,N+1}, v_{2,N+1}, \lambda_{c,N+1} = \lambda_c \quad (t = T_{N+1}) \quad (21)$$

at the communication time point T_{N+1} .

Remark on the condition of the linear system: To improve the condition number of the linear equation system (19), it is useful to scale the constraint equations with the communication step size H . Then, the coupling conditions (4)(c) are replaced by

$$\begin{aligned} \bar{g}_{c\lambda} &:= \frac{x_2 - x_1}{H^2} = 0, \\ \bar{g}'_{c\lambda} &:= \frac{v_2 - v_1}{H} = 0, \\ \bar{g}''_{c\lambda} &:= \dot{v}_2 - \dot{v}_1 = 0. \end{aligned} \quad (22)$$

A detailed analysis, which is not presented here, shows that the condition number can further be improved by using optimized parameters $\alpha_{i,j}$ in Eq. (7). For the case of quadratic extrapolation ($k = 2$), the following optimized basis polynomials can be derived:

$$\begin{aligned} p_1(\tau) &= 12 \cdot (1 - 4 \cdot \tau + 3 \cdot \tau^2), \\ p_2(\tau) &= -6 \cdot (1 - 5 \cdot \tau + 4 \cdot \tau^2), \\ p_3(\tau) &= 1 - 6 \cdot \tau + 6 \cdot \tau^2. \end{aligned} \quad (23)$$

Using cubic polynomials ($k = 3$), the optimized polynomials are given by

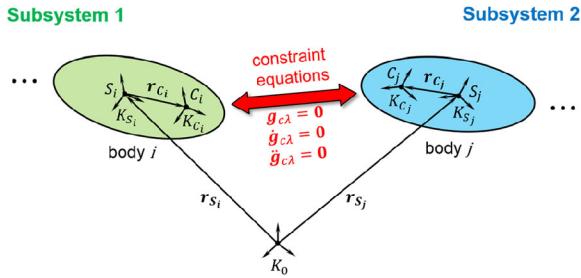
$$\begin{aligned} p_1(\tau) &= 60 \cdot (\tau - 3 \cdot \tau^2 + 2 \cdot \tau^3), \\ p_2(\tau) &= -12 \cdot (2 \cdot \tau - 7 \cdot \tau^2 + 5 \cdot \tau^3), \\ p_3(\tau) &= 3 \cdot \tau - 12 \cdot \tau^2 + 10 \cdot \tau^3. \end{aligned} \quad (24)$$

3 Index-1 co-simulation approach for coupling general mechanical systems

3.1 Problem description

Now, the index-1 co-simulation scheme is described for the case where two arbitrary mechanical subsystems are coupled. Let n_1 be the number of rigid bodies in subsystem 1, and n_2 the corresponding number in subsystem 2. Since the two subsystems are assumed to be general multibody systems, each subsystem is mathematically described by a DAE system. The two DAE systems are coupled by algebraic constraint equations. The configuration is

Fig. 4 Coupling configuration: body i (subsystem 1) coupled to body j (subsystem 2) by means of algebraic constraint equations



illustrated in Fig. 4. The markers K_{C_i} and K_{C_j} are connected by an arbitrary rigid joint, where K_{C_i} belongs to subsystem 1, and K_{C_j} to subsystem 2.

In Fig. 4, the inertia frame of reference is denoted by K_0 . The point S_i terms the center of mass of body i , and \mathbf{r}_{S_i} represents the position vector to S_i . At the center of mass, the body fixed principal axis system K_{S_i} is fixed. To describe the coupling point C_i , the body fixed vector \mathbf{r}_{C_i} is used. At C_i , the body fixed system K_{C_i} is attached, which is assumed to be parallel to K_{S_i} . The notation for subsystem 2 is equivalent.

Let $\mathbf{b} \in \mathbb{R}^3$ be an arbitrary vector. Its coordinates with respect to system K_m are denoted by ${}^{nm}\mathbf{b}$. By means of the rotation matrix ${}^{nm}\mathbf{R} \in \mathbb{R}^{3 \times 3}$, the vector coordinates can be transformed from system K_m to system K_n using the relationship ${}^n\mathbf{b} = {}^{nm}\mathbf{R}{}^m\mathbf{b}$.

To describe the two multibody systems, absolute coordinates are used within this paper [16]. The three coordinates of the center of mass ${}^0\mathbf{r}_{S_q}$ define the translation of an arbitrary rigid body q ; the time derivative ${}^0\mathbf{v}_{S_q} = {}^0\dot{\mathbf{r}}_{S_q}$ describes the velocity of the center of mass S_q . Furthermore, three rotation parameters $\boldsymbol{\gamma}_q \in \mathbb{R}^3$ (e.g., three Euler angles) are used to specify the orientation of the body. The general expression $\dot{\boldsymbol{\gamma}}_q = \mathbf{B}(\boldsymbol{\gamma}_q)^q \boldsymbol{\omega}_q$ with $\mathbf{B}(\boldsymbol{\gamma}_q) \in \mathbb{R}^{3 \times 3}$ relates the coordinates of the angular velocity with respect to the body fixed system K_{S_q} to the time derivative of the rotation parameters.

To keep the notation simple, some auxiliary vectors are defined for the following analysis. Position and velocity coordinates of the two coupling bodies i and j are arranged in the vectors $\mathbf{z}_i = (z_{i1} \ z_{i2} \ \dots \ z_{i12})^T = ({}^0\mathbf{r}_{S_i}^T \ \boldsymbol{\gamma}_i^T \ {}^0\mathbf{v}_{S_i}^T \ {}^i\boldsymbol{\omega}_i^T)^T \in \mathbb{R}^{12}$ and $\mathbf{z}_j = (z_{j1} \ z_{j2} \ \dots \ z_{j12})^T = ({}^0\mathbf{r}_{S_j}^T \ \boldsymbol{\gamma}_j^T \ {}^0\mathbf{v}_{S_j}^T \ {}^j\boldsymbol{\omega}_j^T)^T \in \mathbb{R}^{12}$. The vector $\mathbf{z}_c = (z_i^T \ z_j^T)^T \in \mathbb{R}^{24}$ combines the position and velocity coordinates of both coupling bodies. The position and velocity coordinates of all bodies of subsystem 1 are collected in the vector $\hat{\mathbf{z}}_1 \in \mathbb{R}^{12 \cdot n_1}$; the vector $\hat{\mathbf{z}}_2 \in \mathbb{R}^{12 \cdot n_2}$ collects the position and velocity coordinates of all bodies of subsystem 2. The equations of motion for the coupling bodies, i.e., Newton's and Euler's law, read as follows.

Coupling body i (subsystem 1):

$$\begin{aligned} {}^0\dot{\mathbf{r}}_{S_i} &= {}^0\mathbf{v}_{S_i}, \\ m_i {}^0\dot{\mathbf{v}}_{S_i} &= {}^0\mathbf{F}_{a_i}(\hat{\mathbf{z}}_1, t) + {}^0\mathbf{F}_{r_i} + {}^0\mathbf{F}_{c_i}(\mathbf{z}_c), \\ \dot{\boldsymbol{\gamma}}_i &= \mathbf{B}(\boldsymbol{\gamma}_i)^i \boldsymbol{\omega}_i, \\ {}^i\mathbf{J}_i {}^i\dot{\boldsymbol{\omega}}_i + {}^i\boldsymbol{\omega}_i \times {}^i\mathbf{J}_i {}^i\boldsymbol{\omega}_i &= {}^i\mathbf{M}_{a_i}(\hat{\mathbf{z}}_1, t) + {}^i\mathbf{M}_{r_i} + {}^i\mathbf{M}_{c_i}(\mathbf{z}_c), \end{aligned} \quad (25)$$

Coupling body j (subsystem 2):

$$\begin{aligned} {}^0\dot{\mathbf{r}}_{S_j} &= {}^0\mathbf{v}_{S_j}, \\ m_j {}^0\dot{\mathbf{v}}_{S_j} &= {}^0\mathbf{F}_{a_j}(\hat{\mathbf{z}}_2, t) + {}^0\mathbf{F}_{r_j} + {}^0\mathbf{F}_{c_j}(\mathbf{z}_c), \\ \dot{\boldsymbol{\gamma}}_j &= \mathbf{B}(\boldsymbol{\gamma}_j)^j \boldsymbol{\omega}_j, \\ {}^j\mathbf{J}_j {}^j\dot{\boldsymbol{\omega}}_j + {}^j\boldsymbol{\omega}_j \times {}^j\mathbf{J}_j {}^j\boldsymbol{\omega}_j &= {}^j\mathbf{M}_{a_j}(\hat{\mathbf{z}}_2, t) + {}^j\mathbf{M}_{r_j} + {}^j\mathbf{M}_{c_j}(\mathbf{z}_c). \end{aligned} \quad (26)$$

Here m_i is the mass of the coupling body i ; the matrix ${}^i\mathbf{J}_i$ contains the coordinates of the inertia tensor with respect to K_{S_i} . The vectors ${}^0\mathbf{F}_{r_i}$ and ${}^i\mathbf{M}_{r_i}$ are the reaction forces and torques, which result from algebraic constraints in subsystem 1. The vectors ${}^0\mathbf{F}_{c_i}$ and ${}^i\mathbf{M}_{c_i}$ denote the coupling forces and coupling torques, generated by the coupling constraints. The vector ${}^0\mathbf{F}_{a_i}$ terms the externally applied forces (represented in K_0), and the vector ${}^i\mathbf{M}_{a_i}$ denotes the applied torques (represented in K_{S_i}). For the coupling body j , an equivalent notation is used.

3.2 Coupling of subsystems by rigid joints

By combining three fundamental joints—namely the *atpoint*, the *inplane*, and the *perpendicular joint*—many important other joints can be defined (e.g., the *revolute joint* by one atpoint joint and two perpendicular joints). In [32] the coupling equations for the three fundamental joints are discussed. Here, we discuss the general holonomic and nonscleronomous constraints

$$\mathbf{g}_{c\lambda}(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j) = \mathbf{0} \quad (27)$$

between coupling body i (subsystem 1) and coupling body j (subsystem 2). The coupling forces and coupling torques (see Eqs. (25) and (26)) are

$$\begin{aligned} {}^0\mathbf{F}_{c_i} &= -\mathbf{G}_{c_i}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c, \\ {}^i\mathbf{M}_{c_i} &= -\mathbf{K}_{c_i}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c, \\ {}^0\mathbf{F}_{c_j} &= -\mathbf{G}_{c_j}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c, \\ {}^j\mathbf{M}_{c_j} &= -\mathbf{K}_{c_j}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c, \end{aligned} \quad (28)$$

with matrices

$$\begin{aligned} \mathbf{G}_{c_i}(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j) &= \frac{\partial \mathbf{g}_{c\lambda}}{\partial {}^0\mathbf{r}_{S_i}}, \\ \mathbf{K}_{c_i}(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j) &= \frac{\partial \mathbf{g}_{c\lambda}}{\partial \boldsymbol{\gamma}_i} \mathbf{B}(\boldsymbol{\gamma}_i), \\ \mathbf{G}_{c_j}(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j) &= \frac{\partial \mathbf{g}_{c\lambda}}{\partial {}^0\mathbf{r}_{S_j}}, \\ \mathbf{K}_{c_j}(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j) &= \frac{\partial \mathbf{g}_{c\lambda}}{\partial \boldsymbol{\gamma}_j} \mathbf{B}(\boldsymbol{\gamma}_j). \end{aligned} \quad (29)$$

The vector $\boldsymbol{\lambda}_c$ contains the *Lagrange* multipliers. For the index-1 coupling algorithm, also the first and second time derivatives of Eq. (27) are required.

For the decomposed subsystems, the coupling forces/torques are given by

$$\begin{aligned} {}^0\mathbf{F}_{c_i} &= -\mathbf{G}_{C_i}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\tilde{\mathbf{r}}_{S_j}, \tilde{\boldsymbol{\gamma}}_j)\boldsymbol{\lambda}_c, & {}^i\mathbf{M}_{c_i} &= -\mathbf{K}_{C_i}^T(t, {}^0\mathbf{r}_{S_i}, \boldsymbol{\gamma}_i, {}^0\tilde{\mathbf{r}}_{S_j}, \tilde{\boldsymbol{\gamma}}_j)\boldsymbol{\lambda}_c, \\ {}^0\mathbf{F}_{c_j} &= -\mathbf{G}_{C_j}^T(t, {}^0\tilde{\mathbf{r}}_{S_i}, \tilde{\boldsymbol{\gamma}}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c, & {}^j\mathbf{M}_{c_j} &= -\mathbf{K}_{C_j}^T(t, {}^0\tilde{\mathbf{r}}_{S_i}, \tilde{\boldsymbol{\gamma}}_i, {}^0\mathbf{r}_{S_j}, \boldsymbol{\gamma}_j)\boldsymbol{\lambda}_c \end{aligned} \quad (30)$$

with coupling conditions

$$\begin{aligned}\mathbf{g}_{c\lambda} &:= \mathbf{0}, \\ \mathbf{g}_{cr_i} &:= {}^0\tilde{\mathbf{r}}_{S_i} - {}^0\mathbf{r}_{S_i} = \mathbf{0}, \\ \mathbf{g}_{cy_i} &:= \tilde{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i = \mathbf{0}, \\ \mathbf{g}_{cr_j} &:= {}^0\tilde{\mathbf{r}}_{S_j} - {}^0\mathbf{r}_{S_j} = \mathbf{0}, \\ \mathbf{g}_{cy_j} &:= \tilde{\boldsymbol{\gamma}}_j - \boldsymbol{\gamma}_j = \mathbf{0}.\end{aligned}\tag{31}$$

Note that, for several connecting joints, additional coupling variables ${}^0\tilde{\mathbf{r}}_{S_i}$, $\tilde{\boldsymbol{\gamma}}_i$, ${}^0\tilde{\mathbf{r}}_{S_j}$, and $\tilde{\boldsymbol{\gamma}}_j$ have to be defined. As a consequence, additional coupling equations $\mathbf{g}_{cr_i} = \mathbf{0}$, $\mathbf{g}_{cy_i} = \mathbf{0}$, $\mathbf{g}_{cr_j} = \mathbf{0}$, $\mathbf{g}_{cy_j} = \mathbf{0}$ and related time derivatives have to be introduced. Using an atpoint (spherical) joint, additional coupling variables and corresponding coupling equations are not required. Connecting the two subsystems by an inplane or perpendicular joint, a definition of certain additional coupling variables and coupling equations is necessary (see [32, 34] for further details).

For the subsequent analysis, it is useful to collect the coupling (input) variables $\tilde{\mathbf{u}}_j$ for subsystem 1 and the corresponding coupling (input) variables $\tilde{\mathbf{u}}_i$ for subsystem 2 according to

$$\begin{aligned}\tilde{\mathbf{u}}_j &= (\boldsymbol{\lambda}_c^T \quad {}^0\tilde{\mathbf{r}}_{S_j}^T \quad \tilde{\boldsymbol{\gamma}}_j^T)^T, \\ \tilde{\mathbf{u}}_i &= (\boldsymbol{\lambda}_c^T \quad {}^0\tilde{\mathbf{r}}_{S_i}^T \quad \tilde{\boldsymbol{\gamma}}_i^T)^T.\end{aligned}\tag{32}$$

3.3 General coupling algorithm

The index-1 co-simulation approach introduced in Sect. 2.2 is now generalized for the case of two arbitrary coupled mechanical subsystems. Therefore, we define three different constraint vectors. The vector containing the coupling conditions on position level is denoted by $\mathbf{g}_{c,pos} = (\mathbf{g}_{c\lambda}^T \mathbf{g}_{cz}^T)^T$, where \mathbf{g}_{cz} is a combination of \mathbf{g}_{cr_i} , \mathbf{g}_{cy_i} , \mathbf{g}_{cr_j} , \mathbf{g}_{cy_j} according to the concrete joint used for connecting two coupling bodies. The vector collecting the coupling conditions on velocity level is denoted by $\mathbf{g}_{c,vel} = \dot{\mathbf{g}}_{c,pos}$. Finally, the vector containing the coupling conditions on acceleration level is represented by $\mathbf{g}_{c,acc} = \ddot{\mathbf{g}}_{c,vel}$. The coupling conditions are collected in the resultant collocated coupling vector $\mathbf{g}_c = (\mathbf{g}_{c,pos}^T \mathbf{g}_{c,vel}^T \mathbf{g}_{c,acc}^T)^T$.

As in Sect. 2, we consider the general macro time step from T_N to T_{N+1} to explain the coupling procedure. At the communication time point T_N , the state variables of the subsystems and the coupling variables are given by

$$\begin{aligned}\hat{\mathbf{z}}_1(t = T_N) &= \hat{\mathbf{z}}_{1,N}, & \hat{\mathbf{z}}_2(t = T_N) &= \hat{\mathbf{z}}_{2,N} & \text{(a),} \\ \tilde{\mathbf{u}}_j(t = T_N) &= \tilde{\mathbf{u}}_{j,N}, & \tilde{\mathbf{u}}_i(t = T_N) &= \tilde{\mathbf{u}}_{i,N} & \text{(b).}\end{aligned}\tag{33}$$

For the subsystem integration, the coupling variables $\tilde{\mathbf{u}}_j(t)$ and $\tilde{\mathbf{u}}_i(t)$ have to be approximated. Therefore, the approximation polynomials of degrees $k = 2$ and $k = 3$ introduced in Sect. 2 are used. To enforce the coupling conditions on position, velocity, and acceleration level simultaneously, the approximation polynomial for each coupling variable requires three degrees of freedom, represented by three polynomial parameters. For coupling body i ,

the polynomial parameters are collected in the vector $\boldsymbol{\beta}_{j,N+1} = [\boldsymbol{\beta}_{j1,N+1}^T \ \boldsymbol{\beta}_{j2,N+1}^T \ \boldsymbol{\beta}_{j3,N+1}^T]^T$, and the corresponding vectors for coupling body j are arranged in the vector $\boldsymbol{\beta}_{i,N+1} = [\boldsymbol{\beta}_{i1,N+1}^T \ \boldsymbol{\beta}_{i2,N+1}^T \ \boldsymbol{\beta}_{i3,N+1}^T]^T$. Hence, we can write the approximation polynomials for the macro time step $T_N \rightarrow T_{N+1}$ as

$$\begin{aligned}\tilde{\mathbf{u}}_j(t) &= \tilde{\mathbf{u}}_{j,N+1}(\boldsymbol{\beta}_{j,N+1}; t) \\ &= \begin{cases} \boldsymbol{\beta}_{j1,N+1} \cdot p_1(\tau) + \boldsymbol{\beta}_{j2,N+1} \cdot p_2(\tau) + \boldsymbol{\beta}_{j3,N+1} \cdot p_3(\tau) & (k=2) \\ \tilde{\mathbf{u}}_{j,N} + \boldsymbol{\beta}_{j1,N+1} \cdot p_1(\tau) + \boldsymbol{\beta}_{j2,N+1} \cdot p_2(\tau) + \boldsymbol{\beta}_{j3,N+1} \cdot p_3(\tau) & (k=3), \end{cases} \\ \tilde{\mathbf{u}}_i(t) &= \tilde{\mathbf{u}}_{i,N+1}(\boldsymbol{\beta}_{i,N+1}; t) \\ &= \begin{cases} \boldsymbol{\beta}_{i1,N+1} \cdot p_1(\tau) + \boldsymbol{\beta}_{i2,N+1} \cdot p_2(\tau) + \boldsymbol{\beta}_{i3,N+1} \cdot p_3(\tau) & (k=2) \\ \tilde{\mathbf{u}}_{i,N} + \boldsymbol{\beta}_{i1,N+1} \cdot p_1(\tau) + \boldsymbol{\beta}_{i2,N+1} \cdot p_2(\tau) + \boldsymbol{\beta}_{i3,N+1} \cdot p_3(\tau) & (k=3) \end{cases} \\ &\quad \left(\tau = \frac{t - T_N}{T_{N+1} - T_N} \right).\end{aligned}\tag{34}$$

The polynomials $p_1(\tau)$, $p_2(\tau)$, and $p_3(\tau)$ are defined in Eq. (7). The basic idea of the linear implicit co-simulation scheme has already been presented in Sect. 2 for the co-simulation test model. Therefore, we only sketch the three integration steps for the general case.

Step 1: Predictor step

- The predicted coupling variables $\boldsymbol{\beta}_{c,N+1}^p$ are generated by the approximation polynomials of the previous communication time step (note that the vector $\boldsymbol{\beta}_{c,N+1}^p$ collects the polynomial parameters of both subsystems). Integration of the subsystems yields the predicted state variables. The corresponding accelerations are calculated with the help of the equations of motion. Finally, the residuals

$$\mathbf{g}_{c,N+1}^p := \mathbf{g}_{c,N+1}(\boldsymbol{\beta}_{c,N+1}^p)\tag{35}$$

are computed.

Step 2: Calculation of corrected coupling variables

- Within the linear implicit approach, only the linearized coupling conditions

$$\mathbf{g}_{c,N+1}^{linear}(\boldsymbol{\beta}_{c,N+1}^*) := \mathbf{g}_{c,N+1}^p + \mathbf{J}_{N+1} \cdot (\boldsymbol{\beta}_{c,N+1}^* - \boldsymbol{\beta}_{c,N+1}^p) = \mathbf{0}\tag{36}$$

are considered, where $\boldsymbol{\beta}_{c,N+1}^*$ represents general polynomial parameters.

- The coupling *Jacobian* matrix can generally be expressed as

$$\mathbf{J}_{N+1} = \frac{\partial \mathbf{g}_{c,N+1}}{\partial \boldsymbol{\beta}_{c,N+1}^*} \Big|_{\boldsymbol{\beta}_{c,N+1}^p} = \frac{\partial \mathbf{g}_{c,N+1}}{\partial \mathbf{z}_{c,N+1}^*} \frac{\partial \mathbf{z}_{c,N+1}^*}{\partial \boldsymbol{\beta}_{c,N+1}^*} \Big|_{\boldsymbol{\beta}_{c,N+1}^p}.\tag{37}$$

Regardless of the complexity of the model and the constraint equations, calculation of the *Jacobian* only requires the calculation of the partial derivatives $\frac{\partial \mathbf{z}_{c,N+1}^*}{\partial \boldsymbol{\beta}_{c,N+1}^*} \Big|_{\boldsymbol{\beta}_{c,N+1}^p}$, since $\frac{\partial \mathbf{g}_{c,N+1}}{\partial \mathbf{z}_{c,N+1}^*}$ can be simply calculated analytically for arbitrary nonscleronomous constraints (for

nonholonomic constraints, the approach has to be modified). In general, we can compute these partial derivatives numerically by approximating the partial derivatives by finite differences. Therefore, the predictor step has to be repeated with perturbed variables $\boldsymbol{\beta}_{c,N+1}^p + \Delta\boldsymbol{\beta}$ (see [34] for more detail). Note that we can carry out the simulations with the perturbed parameters in parallel with the predictor step so that almost no additional simulation time will be required for calculating the *Jacobian* matrix. If a large number of subsystems are coupled, a parallel calculation of the *Jacobian* matrix may become difficult or even impossible. In special cases, for instance, for linear subsystems, alternative methods may be used for computing the coupling *Jacobian* \mathbf{J}_{N+1} . If the *Jacobian* matrices of the subsystems are known, then these matrices might also be used for approximating the coupling *Jacobian*.

- By solving Eq. (36) for $\boldsymbol{\beta}_{c,N+1}^*$ we can obtain improved coupling variables. This yields the corrected polynomial parameters

$$\boldsymbol{\beta}_{c,N+1} = \boldsymbol{\beta}_{c,N+1}^p - \mathbf{J}_{N+1}^{-1} \cdot \mathbf{g}_{c,N+1}(\boldsymbol{\beta}_{c,N+1}^p). \quad (38)$$

Step 3: Corrector step

- Using interpolation polynomials with the corrected parameters of Eq. (38), a subsystem integration yields the corrected state and coupling variables at T_{N+1} .

4 Numerical examples

4.1 Linear 1-DOF oscillator

We consider again the linear 1-DOF oscillator of Sect. 2. Simulations have been accomplished with the following parameters: $m_1 = 1$ kg, $m_2 = 2$ kg, $c = c_1 = c_2 = 1000$ N/m, and $d = d_1 = d_2 = 10$ Ns/m. As initial conditions, we have set $x_{1,0} = x_{2,0} = 0$ m, $v_{1,0} = v_{2,0} = 100$ m/s, and $\lambda_{c,0} = 1000/3$ N. The two subsystems have been integrated analytically. Figure 5 shows the time response $x_1(t)$ of the mass 1, the velocity $v_1(t)$, and the *Lagrange* multiplier $\lambda_c(t)$ for the index-1 coupling approach for $k = 2$ and $k = 3$. The simulations have been carried out with the macro step size $H = 1E - 2$ s. As can be seen, the two co-simulation approaches agree very well with the analytical reference solution. The error in the amplitudes is very small; a phase shift is not detected. The zoom plots indicate that the simulation with cubic approximation polynomials is little bit closer to the reference than the solution with quadratic polynomials. Figure 6 depicts the residuals $x_2 - x_1$, $v_2 - v_1$, and $\dot{v}_2 - \dot{v}_1$ for $k = 2$ and $k = 3$. The residuals on position, velocity, and acceleration level are almost zero, although the macro step size is rather large. In this context, it should be noticed that the test model is linear, i.e., within a predictor/corrector co-simulation approach, only one *Newton*-step is required to obtain corrected coupling variables. Considering nonlinear models in connection with a linear implicit co-simulation approach, additional numerical errors are introduced if only one *Newton* step is carried out. This may also explain the small numerical errors observed for the linear test model.

To get a more detailed view on the quantitative numerical error generated by the co-simulation, the global and local errors are calculated for different macro step sizes H . The global errors $\varepsilon_{pos,glo}$, $\varepsilon_{vel,glo}$ of the position and velocity variables are plotted in Fig. 7 as functions of H . As a reference solution, we used the analytical solution of Eq. (1). We can see that, for quadratic approximation, the global errors converge with $\mathcal{O}(H^5)$. In the case of cubic approximation, the global errors converge with $\mathcal{O}(H^6)$.

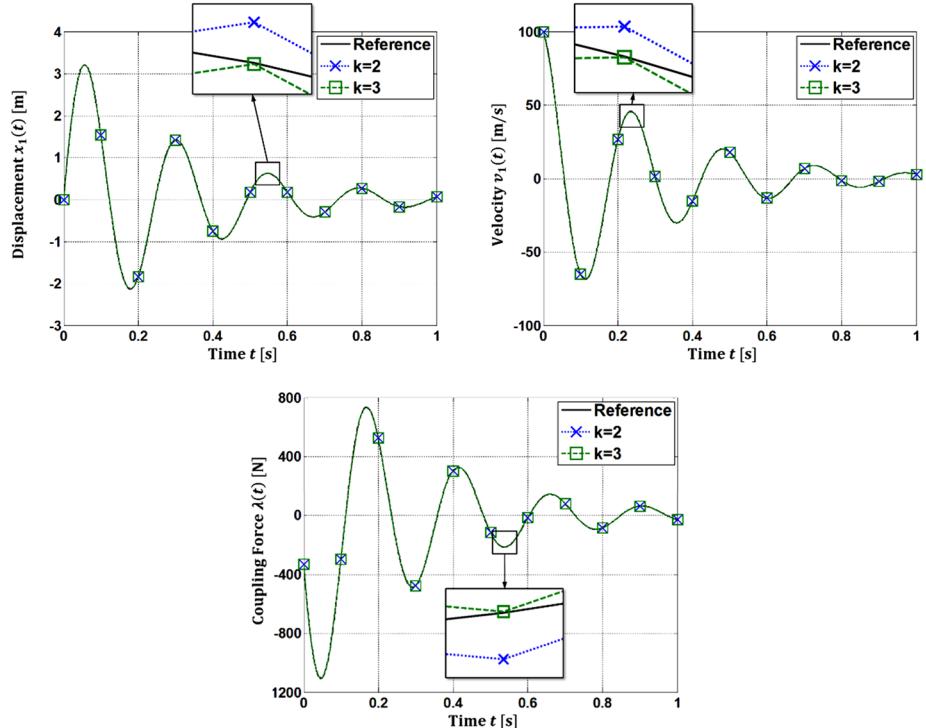


Fig. 5 Displacement $x_1(t)$, velocity $v_1(t)$, and Lagrange multiplier $\lambda_c(t)$ for the index-1 co-simulation approach ($k = 2$ and $k = 3$, $H = 1E - 2$)

The corresponding local errors are depicted in Fig. 8. We can observe that, for quadratic approximation, the local errors converge with $\mathcal{O}(H^6)$. For cubic approximation, the local errors converge with $\mathcal{O}(H^7)$.

The small errors, even for larger macro step sizes, are surprising. It should, however, be stressed that the subsystems have been integrated analytically. Consequently, the error plots only contain the error introduced by the co-simulation procedure.

4.2 Double pendulum

Next, we investigate the planar double pendulum (masses $m_1 = m_2 = 1$ kg, link lengths $\ell_1 = \ell_2 = 1$ m) illustrated in Fig. 9. Link 1 (subsystem 1) is connected to ground by a revolute joint. Link 2 (subsystem 2) is coupled to link 1 by an atpoint joint, which is in the planar case equivalent to a revolute joint. Gravity is acting in negative y -direction ($g = 9.81$ m/s 2). As initial conditions, we have set $\varphi_{1,0} = \frac{\pi}{4}$, $\varphi_{2,0} = -\frac{\pi}{4}$, $\dot{\varphi}_{1,0} = \dot{\varphi}_{2,0} = 0$ s $^{-1}$. We accomplished the simulations with a constant macro step size $H = 1E - 2$ s. We used the linear-implicit third-fourth-order Rosenbrock–Wanner method (variable step size with error tolerance $1E - 23$) for integrating the subsystems, namely the rosenbrock_dae solver from Maple [23]. Figure 10 illustrates the angles $\varphi_1(t)$ and $\varphi_2(t)$. We computed the reference solution numerically with a monolithic model. Although the macro step size is rather large, the plots clearly indicate that the co-simulation yields very accurate results with respect to both amplitudes and phase angle. Figure 11 shows the residuals $g_{ca_x}(t) = x_1 + \frac{l_1}{2} \cdot \cos \varphi_1 +$

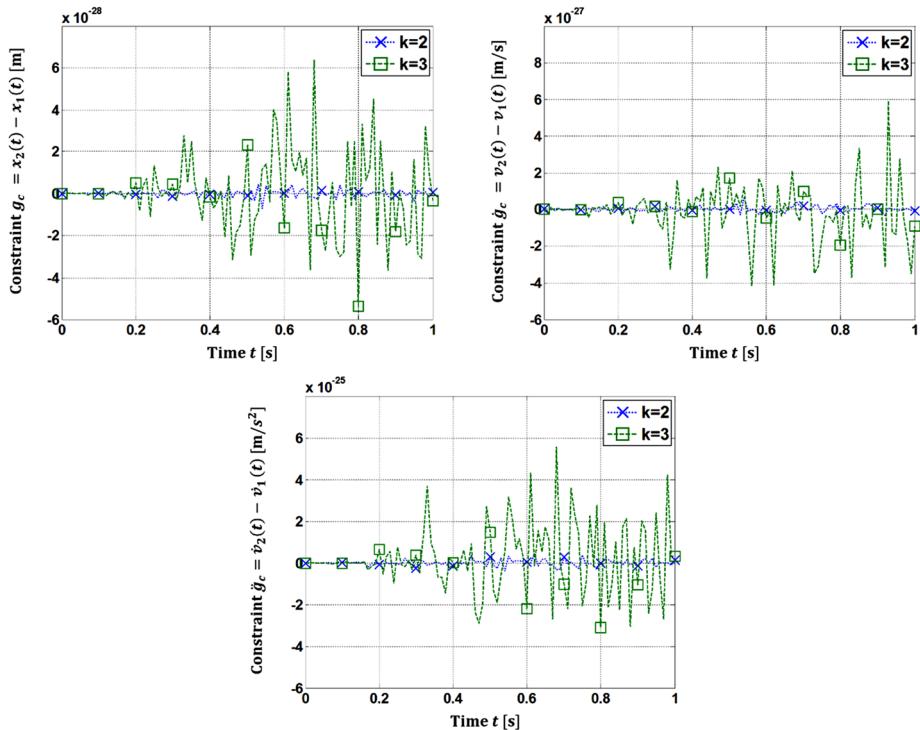


Fig. 6 Residuals $x_2 - x_1$, $v_2 - v_1$, and $\dot{v}_2 - \dot{v}_1$ for the index-1 co-simulation approach ($k = 2$ and $k = 3$, $H = 1E - 2$)

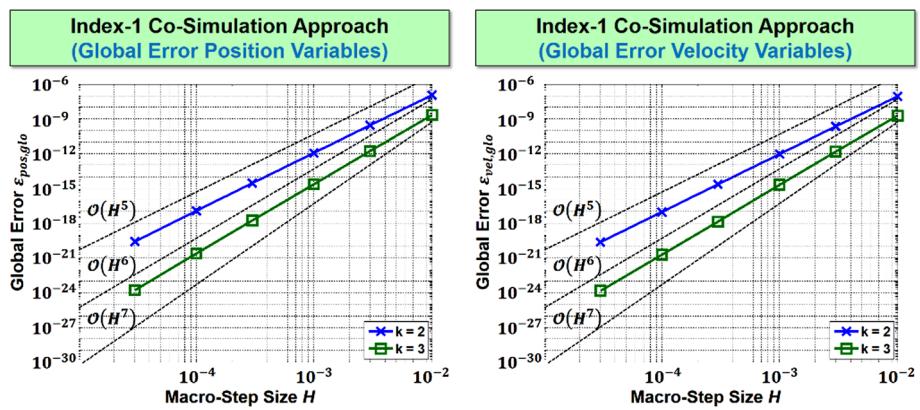


Fig. 7 Convergence plots for the 1-DOF oscillator: Global error over the macro step size H for the index-1 co-simulation approach ($k = 2$ and $k = 3$)

$\frac{l_2}{2} \cdot \cos \varphi_2 - x_2$ and $g_{cay}(t) = y_1 + \frac{l_1}{2} \cdot \sin \varphi_1 + \frac{l_2}{2} \cdot \sin \varphi_2 - y_2$ of the constraint equations on position level and the corresponding residuals on velocity and acceleration level. Compared to the linear test model of Sect. 4.1, larger residuals are detected. The reason therefore is that for a nonlinear model, the coupling conditions are not exactly fulfilled at the macro time

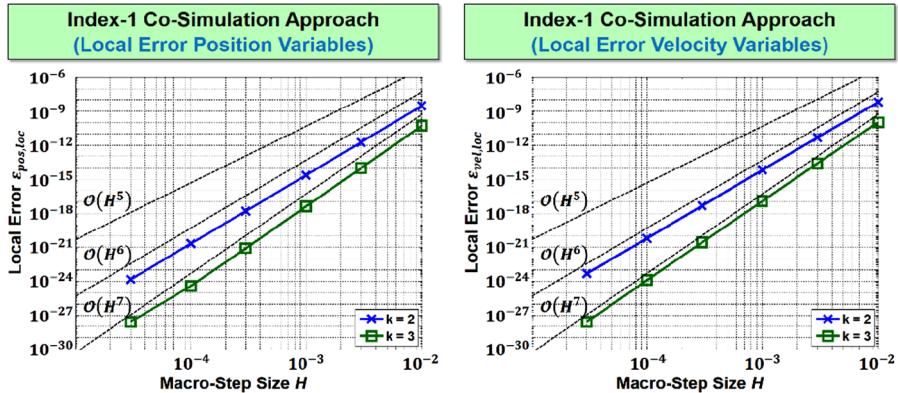


Fig. 8 Convergence plots for the 1-DOF oscillator: Local error over the macro step size H for the index-1 co-simulation approach ($k = 2$ and $k = 3$)

Fig. 9 Double pendulum:
Interpretation as two links
coupled by an atpoint joint

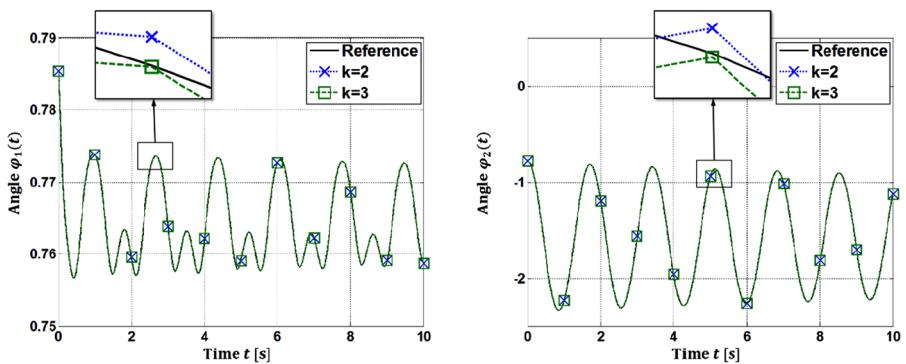
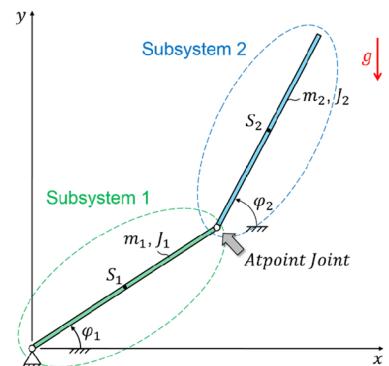


Fig. 10 Co-simulation results for the double pendulum: Angles $\varphi_1(t)$ and $\varphi_2(t)$ ($H = 1E - 2$)

points; due to the linear implicit implementation, only the linearized coupling conditions are enforced. This may also explain the spikes in Fig. 11 for $k = 2$.

The global errors $\varepsilon_{pos,glo}$ and $\varepsilon_{vel,glo}$ of the position and velocity variables are plotted in Fig. 12 as functions of the macro step size. The subsystem integrations have been accom-

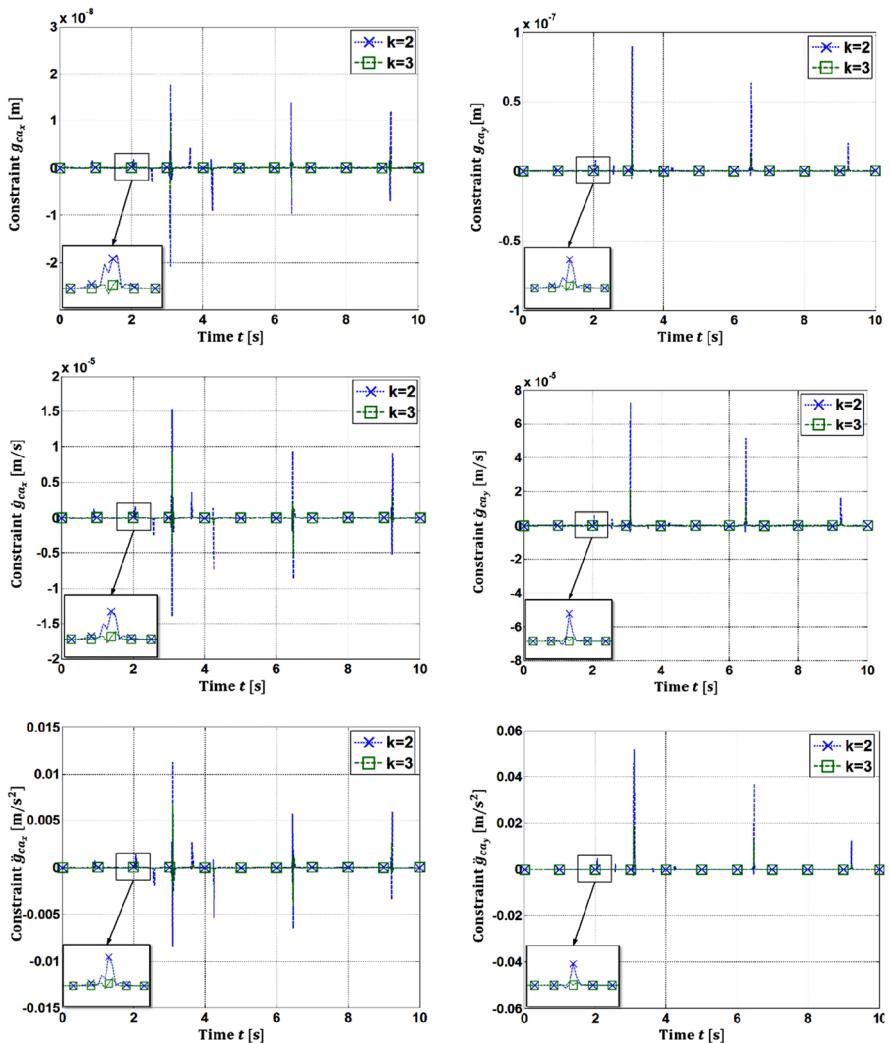


Fig. 11 Co-simulation results for the double pendulum: Residuals g_{ca_x} , g_{ca_y} and their first and second derivatives ($H = 1E - 2$)

plished very precisely with a linear-implicit third-fourth-order Rosenbrock–Wanner method using an error tolerance of $1E - 23$. To achieve accurate results, we carried out all calculations with 31 decimal places. Compared with the results of Sect. 4.1, we observed the same convergence behavior: The global errors converge with $\mathcal{O}(H^5)$ for $k = 2$ and $\mathcal{O}(H^6)$ for $k = 3$, and the corresponding local errors converge with $\mathcal{O}(H^6)$ and $\mathcal{O}(H^7)$ see Fig. 13.

5 Conclusions

We presented a solver coupling scheme, which is tailored to couple mechanical subsystems by arbitrary (also nonscleronic) algebraic constraint equations. Here, we only considered

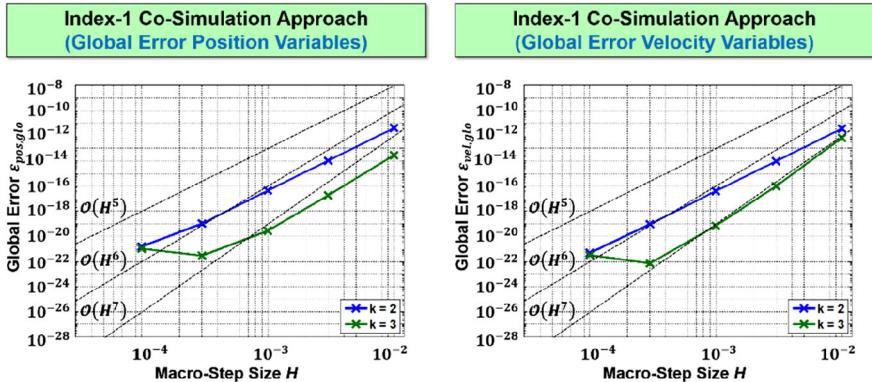


Fig. 12 Convergence plots for the double pendulum: Global error over the macro step size H for the index-1 co-simulation approach ($k = 2$ and $k = 3$)

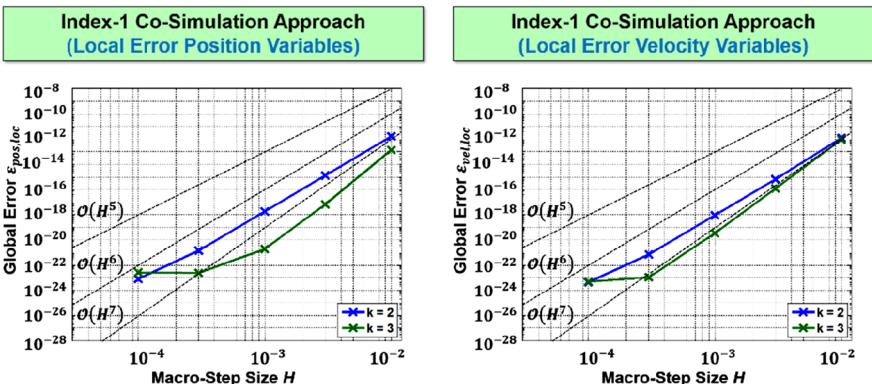


Fig. 13 Convergence plots for the double pendulum: Local error over the macro step size H for the index-1 co-simulation approach ($k = 2$ and $k = 3$)

holonomic constraints; for nonholonomic constraints, the approach has to be modified. The method can be applied either to parallelize dynamical systems or to couple different solvers in the framework of a co-simulation. We showed that the algorithm yields very accurate and very stable results, since the constraints and also the hidden constraints are taken into account. We have discussed two cases: approximation of the coupling variables with quadratic polynomials, which are discontinuous at the communication time points, and a cubic approximation technique, which entails continuous coupling variables at the communication time points. As has been shown, the first approach shows a local convergence behavior of $\mathcal{O}(H^6)$ for the position, velocity, and acceleration variables. The second approach converges with $\mathcal{O}(H^7)$. We carried out a stability analysis for both cases. Both approaches exhibit a good stability behavior, which outperforms the numerical stability of recently examined methods by the authors [32, 34, 36].

Although convergence and stability of the presented approach are very promising, the method has two major drawbacks. The first main disadvantage is that the method requires a coupling *Jacobian* matrix. Without access to the subsystem solvers, the coupling *Jacobian* has to be calculated numerically, for instance, by finite differences. If only two subsystems

are coupled, then the dimension of the coupling *Jacobian* is small, so that the *Jacobian* may be calculated in parallel with the predictor step without a significant increase of simulation time. If multiple subsystems are coupled, then a parallel calculation may become problematic or impossible. A second drawback of the presented method is that a multirate factor larger than 3 has to be introduced to correctly calculate the corrected coupling polynomials.

In future work, the method should be tested for large-scale mechanical problems. Because of high convergence order of the method, the approach might be a practically interesting alternative to well-established coupling techniques known from the literature. A second point for future investigations concerns the implementation of nonequidistant communication step sizes. The crucial point for realizing variable macro steps lies in the derivation of an error estimator for the local error introduced by the co-simulation. From the mathematical point of view, developing an appropriate error estimator is an interesting but nontrivial problem.

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

Appendix A: Numerical stability of the index-1 co-simulation approach

The numerical stability of time integration schemes is defined by Dahlquist's test equation $\dot{y}(t) = \Lambda \cdot y(t)$ with complex constant $\Lambda = \Lambda_r + i\Lambda_i$. This test equation can be interpreted as the complex representation of the homogenous linear single-mass oscillator. Dahlquist's test equation cannot directly be used for analyzing the stability of co-simulation methods. For analyzing the stability of co-simulation methods, a proper test model is therefore required. The straightforward generalization of Dahlquist's test model for analyzing the stability of co-simulation methods is the two-mass oscillator of Sect. 2, which represents two algebraically coupled Dahlquist equations. The two-mass oscillator is described by five independent parameters and may be considered as a general (complete) linear test model for analyzing the numerical stability of coupling methods with algebraic constraints.

For the stability analysis, we introduce dimensionless variables. The dimensionless time is denoted by $\bar{t} = \frac{t}{H}$. The derivative with respect to the dimensionless time \bar{t} is represented by $(\cdot)' = \frac{d(\cdot)}{dt}$. We assume that \bar{x}_1, \bar{x}_2 are properly chosen dimensionless position coordinates. The dimensionless velocities are characterized by $\bar{v}_1 = H \cdot v_1$ and $\bar{v}_2 = H \cdot v_2$. The dimensionless coupling force is termed by $\bar{\lambda}_c = \frac{\lambda_c \cdot H^2}{m_1}$ and represented by the polynomial $\bar{\lambda}_c(\bar{t}) = \bar{\lambda}_{c,N+1}(\bar{\beta}_{N+1}; \bar{t})$ with $\bar{\beta}_{N+1} = \frac{\beta_{N+1} \cdot H^2}{m_1}$. We additionally introduce the following five parameters

$$\bar{c}_1 = \frac{c_1 \cdot H^2}{m_1}, \quad \bar{d}_1 = \frac{d_1 \cdot H}{m_1}, \quad \alpha_m = \frac{m_2}{m_1}, \quad \alpha_c = \frac{c_2}{c_1}, \quad \alpha_d = \frac{d_2}{d_1}. \quad (39)$$

The dimensionless form of the equations of motion for the two-mass oscillator of Sect. 2 then reads as

$$\begin{aligned} \bar{x}'_1 &= \bar{v}_1, \\ \bar{v}'_1 &= -\bar{c}_1 \cdot \bar{x}_1 - \bar{d}_1 \cdot \bar{v}_1 + \bar{\lambda}_c, \\ \bar{x}'_2 &= \bar{v}_2, \end{aligned} \quad (40)$$

$$\bar{v}'_2 = -\frac{\alpha_c}{\alpha_m} \cdot \bar{c}_1 \cdot \bar{x}_2 - \frac{\alpha_d}{\alpha_m} \cdot \bar{d}_1 \cdot \bar{v}_2 - \frac{1}{\alpha_m} \cdot \bar{\lambda}_c,$$

$$0 = \bar{x}_2 - \bar{x}_1.$$

If $m_1, m_2, c_1, c_2, d_1, d_2 > 0$, then the two-mass oscillator is physically stable. In terms of a co-simulation approach, the two-mass oscillator is decomposed into two subsystems, and a homogenous linear system of recurrence equations can be derived, which describes the discrete-time co-simulation results. The co-simulation is called numerically stable if all eigenvalues of the recurrence system are inside the unit circle. The eigenvalues depend on five independent test model parameters specified in Eq. (39).

We analyze the stability of time integration algorithms by discretizing Dahlquist's test equation with a time integration scheme. This yields a linear recurrence equation. The spectral radius ρ of this recurrence equation determines the stability of the time integration method and depends on two parameters $h\Lambda_r$ and $h\Lambda_i$. Hence, we can generate 2D-stability plots, which illustrate the complete stability behavior of the integration scheme. Discretizing the co-simulation test model with a co-simulation scheme yields a linear system of recurrence equations. The spectral radius of this recurrence system depends on five independent test model parameters defined in Eq. (39). Consequently, it is impossible to depict the complete stability behavior with 2D-stability plots. Therefore one possibility is to fix three of the entire five test model parameters and to depict the spectral radius of the recurrence system as a function of two properly chosen parameters.

Bearing in mind the stability definition for numerical time integration schemes, where the two parameters $h\Lambda_r$ and $h\Lambda_i$ are used to characterize the stability behavior, it seems to be advantageous not to use the five co-simulation test model parameters introduced in Eq. (39), but to employ five different parameters. The following five parameters might be more convenient to be used as independent test model parameters:

$$\begin{aligned} \overline{\Lambda}_{r1} &= -\frac{\bar{d}_1}{2}, & \overline{\Lambda}_{i1} &= \frac{1}{2}\sqrt{4 \cdot \bar{c}_1 - \bar{d}_1^2}, \\ \alpha_m &= \frac{m_2}{m_1}, & \alpha_{\Lambda_r} &= \frac{\overline{\Lambda}_{r2}}{\overline{\Lambda}_{r1}} = \frac{\alpha_d}{\alpha_m}, \\ \alpha_{\Lambda_i} &= \frac{\overline{\Lambda}_{i2}}{\overline{\Lambda}_{i1}} = \frac{1}{\alpha_m} \frac{\sqrt{4 \cdot \alpha_m \cdot \alpha_c \cdot \bar{c}_1 - \alpha_d^2 \cdot \bar{d}_1^2}}{\sqrt{4 \cdot \bar{c}_1 - \bar{d}_1^2}}. \end{aligned} \quad (41)$$

The physical interpretation of the new parameters is straightforward: $\overline{\Lambda}_{r1}$ and $\overline{\Lambda}_{i1}$ define subsystem 1 and represent the real and imaginary parts of the eigenvalue of subsystem 1; the mass-ratio is characterized by the coefficient α_m ; The parameters α_{Λ_r} and α_{Λ_i} denote the ratio of the real and imaginary part of subsystem 2 with respect to subsystem 1; they describe the asymmetry of the co-simulation test model and characterize the damping and the frequency ratio of subsystem 2 with respect to subsystem 1. Regarding the stability plots for numerical time integration schemes, it seems to be convenient to use the parameters $\overline{\Lambda}_{r1}$ and $\overline{\Lambda}_{i1}$ as axes for 2D-stability plots and to fix the remaining three parameters. In the subsequent stability analysis, the three parameters α_m , α_{Λ_r} , and α_{Λ_i} are therefore assumed to be constant, and the spectral radius is computed as a function of $\overline{\Lambda}_{r1}$ and $\overline{\Lambda}_{i1}$.

With the dimensionless variables and parameters introduced before, the decomposed system reads as

Subsystem 1:

$$\begin{aligned}\bar{x}'_1 &= \bar{v}_1, \\ \bar{v}'_1 &= (\bar{\Lambda}_{r1}^2 + \bar{\Lambda}_{i1}^2) \cdot \bar{x}_1 + 2 \cdot \bar{\Lambda}_{r1} \cdot \bar{v}_1 + \bar{\lambda}_c\end{aligned}\quad (a),$$

Subsystem 2:

$$\begin{aligned}\bar{x}'_2 &= \bar{v}_2, \\ \bar{v}'_2 &= (\bar{\Lambda}_{r1}^2 \alpha_{Ar}^2 + \bar{\Lambda}_{i1}^2 \alpha_{Ai}^2) \cdot \bar{x}_2 + 2 \cdot \bar{\Lambda}_{r1} \cdot \alpha_{Ar} \cdot \bar{v}_2 - \frac{1}{\alpha_m} \bar{\lambda}_c\end{aligned}\quad (b),$$

Coupling condition:

$$0 = \bar{x}_2 - \bar{x}_1 \quad (c).$$

We denote the dimensionless macro time point by $\bar{T}_N = \frac{T_N}{H}$. At the beginning of the macro time step, the state variables and the coupling variable are assumed to be known:

$$\begin{aligned}\bar{x}_1(\bar{t} = \bar{T}_N) &= \bar{x}_{1,N}, & \bar{v}_1(\bar{t} = \bar{T}_N) &= \bar{v}_{1,N}, \\ \bar{x}_2(\bar{t} = \bar{T}_N) &= \bar{x}_{2,N}, & \bar{v}_2(\bar{t} = \bar{T}_N) &= \bar{v}_{2,N}, \\ \bar{\lambda}_c(\bar{t} = \bar{T}_N) &= \bar{\lambda}_{c,N}.\end{aligned}\quad (a) \quad (b).$$

We define the auxiliary vector $\bar{z}_N = (\bar{x}_{1,N}, \bar{v}_{1,N}, \bar{x}_{2,N}, \bar{v}_{2,N}, \bar{\lambda}_c)^T$, which contains the dimensionless variables of the two subsystems and the dimensionless coupling force at the macro time point \bar{T}_N . Using the general coupling force

$$\bar{\lambda}_c^*(\bar{t}) = \bar{\lambda}_{c,N+1}(\bar{\beta}_{N+1}^*; \bar{t}), \quad (44)$$

an integration of the two subsystems from \bar{T}_N to \bar{T}_{N+1} with the initial conditions (43)(a) yields the general dimensionless variables

$$\bar{x}_{1,N+1}^*, \bar{v}_{1,N+1}^*, \bar{x}_{2,N+1}^*, \bar{v}_{2,N+1}^*, \bar{\lambda}_{c,N+1}^* = \bar{\lambda}_c^* \quad (\bar{t} = \bar{T}_{N+1}) \quad (45)$$

at the communication time point \bar{T}_{N+1} , which are functions of $\bar{\beta}_{N+1}^*$ and \bar{z}_N . Then the accelerations at the communication time point \bar{T}_{N+1} are computed, i.e.,

$$\bar{v}_{1,N+1}^{*\prime}, \bar{v}_{2,N+1}^{*\prime}. \quad (46)$$

Since the dimensionless variables depend linearly on the coefficient vector $\bar{\beta}_{N+1}^*$, three constraint equations

$$\begin{aligned}0 &= \bar{x}_{2,N+1}^*(\bar{\beta}_{N+1}^*; \bar{z}_N) - \bar{x}_{1,N+1}^*(\bar{\beta}_{N+1}^*; \bar{z}_N), \\ 0 &= \bar{v}_{2,N+1}^*(\bar{\beta}_{N+1}^*; \bar{z}_N) - \bar{v}_{1,N+1}^*(\bar{\beta}_{N+1}^*; \bar{z}_N), \\ 0 &= \bar{v}_{2,N+1}^{*\prime}(\bar{\beta}_{N+1}^*; \bar{z}_N) - \bar{v}_{1,N+1}^{*\prime}(\bar{\beta}_{N+1}^*; \bar{z}_N)\end{aligned}\quad (47)$$

can be solved by one *Newton* iteration step. Hence the corrected polynomial coefficients are independent of the predicted coupling force. With the corrected coefficient vector $\bar{\beta}_{N+1}$, we

finally calculate the corrected variables

$$\begin{aligned}\bar{x}_{1,N+1} &= \bar{x}_{1,N+1}(\bar{z}_N), & \bar{v}_{1,N+1} &= \bar{v}_{1,N+1}(\bar{z}_N), \\ \bar{x}_{2,N+1} &= \bar{x}_{2,N+1}(\bar{z}_N), & \bar{v}_{2,N+1} &= \bar{v}_{2,N+1}(\bar{z}_N), \\ \bar{\lambda}_{c,N+1} &= \bar{\lambda}_c \quad (\bar{t} = \bar{T}_{N+1}; \bar{z}_N).\end{aligned}\quad (48)$$

Note that the vector \bar{z}_{N+1} depends linearly on the initial values z_N . Hence we obtain a system of five coupled linear recurrence equations of the form

$$A_{N+1} \cdot \bar{z}_{N+1} + A_N \cdot \bar{z}_N = \mathbf{0}. \quad (49)$$

We study the stability of the homogenous system (49) by an eigenvalue analysis. With the exponential approach $\bar{z}_N = \hat{z}_j \cdot \Lambda_j^N$, we can calculate the eigenvalues Λ_j and the corresponding eigenvectors \hat{z}_j ($j = 1, \dots, 5$). The stability of the recurrence system depends on the magnitude of the largest eigenvalue, which terms the spectral radius $\rho = \max\{|\Lambda_j|\}$. For $\rho \leq 1$, the underlying co-simulation approach is stable, and otherwise unstable.

We should stress that the spectral radius can only be calculated numerically. Hence the three parameters $\alpha_m, \alpha_{Ar}, \alpha_{Ai}$ are fixed, and ρ is plotted for different values of $\bar{\Lambda}_{r1}$ and $\bar{\Lambda}_{i1}$. If $\rho \leq 1$ for a certain parameter combination, then we mark the corresponding point as a

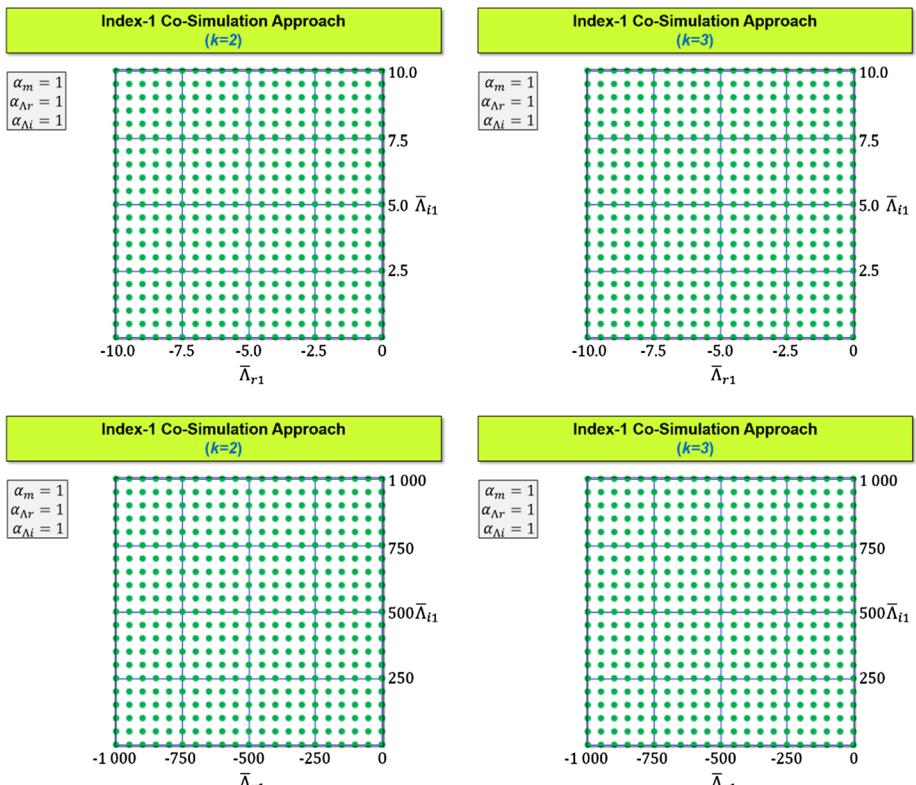


Fig. 14 Stability plots for the symmetrical co-simulation test model: left column for quadratic approximation polynomials ($k = 2$); right column for cubic approximation polynomials ($k = 3$)

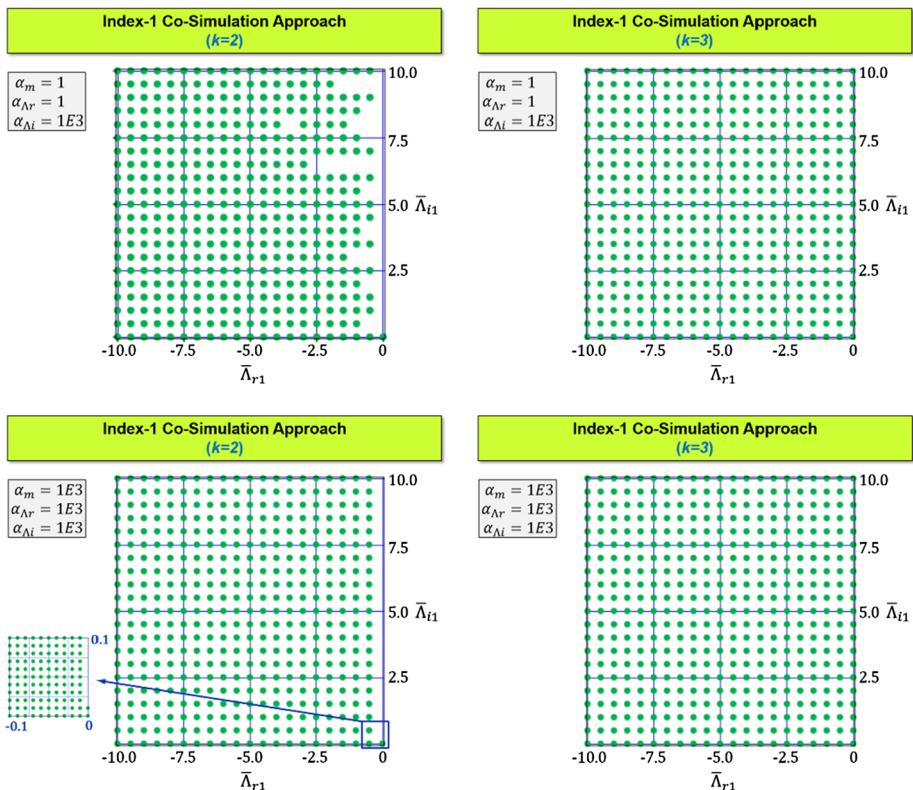


Fig. 15 Stability plots for the asymmetrical co-simulation test model: left column for quadratic approximation polynomials ($k = 2$); right column for cubic approximation polynomials ($k = 3$)

green point in the $(\bar{\Lambda}_{r1}, \bar{\Lambda}_{i1})$ -plane. If $\rho > 1$, then we plot no point in the stability diagram. Since we can calculate ρ only for discrete parameter combinations, stability information is available only at discrete grid points.

For the case of a symmetrical test model ($\alpha_m = \alpha_{Ar} = \alpha_{Ai} = 1$), we collect stability plots in Fig. 14. To get information over a larger parameter region, we use two different scales, namely $[-10, 10]$ and $[-1000, 1000]$. For the symmetrical test model, only stable points are observed, also for the case of large damping values $\bar{\Lambda}_{r1}$ and large stiffnesses $\bar{\Lambda}_{i1}$, both for quadratic ($k = 2$) and cubic ($k = 3$) interpolation polynomials. Further calculations, not shown here, for even larger values of $\bar{\Lambda}_{i1}$ and $\bar{\Lambda}_{r1}$ also yielded only stable points. Hence the numerical stability analysis seems to indicate that the symmetrical test model is stable in the complete left half-plane (comparable with an A-stable time integration scheme). We should however stress again that stability in the left half-plane cannot be shown analytically, since the spectral radius can only be evaluated numerically.

We discuss the influence of asymmetry with the help of Fig. 15. Firstly, only the mass ratio coefficient α_m is changed from 1 to $1E3$. No change in the stability plots is observed, neither for $k = 2$ nor for $k = 3$. Next, the damping ratio coefficient α_{Ar} is increased to $1E3$. Again, we detect only stable points in the diagrams. Since both cases—($\alpha_m = 1E3, \alpha_{Ar} = 1, \alpha_{Ai} = 1$) and ($\alpha_m = 1, \alpha_{Ar} = 1E3, \alpha_{Ai} = 1$)—only yield stable points, the corresponding plots are not shown in Fig. 15. Then, we examine the influence of the frequency ratio. Here

a significant reduction of the numerical stability can be seen for the case $k = 2$, and an unstable region can clearly be observed close to the vertical axes, i.e., for systems with low damping. Interesting, and somewhat surprising, is that, for $k = 3$, no unstable points occur. Finally, we change all three parameters simultaneously ($\alpha_m = \alpha_{Ar} = \alpha_{Ai} = 1E3$). Using cubic polynomials ($k = 3$), all points in the considered parameter range are stable. For $k = 2$, again unstable points are detected close to the \overline{A}_{i1} -axis. The region of instability is a little bit smaller compared to the case $\alpha_m = 1, \alpha_{Ar} = 1, \alpha_{Ai} = 1E3$, most probably due to the increased damping introduced by the parameter $\alpha_{Ar} = 1E3$.

Summary: Using cubic approximation polynomials, the proposed co-simulation approach exhibits a very stable stability behavior. In the case where the subsystems are very different ($\alpha_m \neq 1, \alpha_{Ar} \neq 1, \alpha_{Ai} \neq 1$), we are often faced with severe numerical stability problems, especially if higher-order approximation polynomials are used [5, 11, 15, 17, 34, 35]. The method discussed here provides stable results, even for critical parameter values and for the practically interesting case where the subsystems are integrated in parallel (*Jacobi*-type).

References

- Alioli, M., Morandini, M., Masarati, P.: Coupled multibody-fluid dynamics simulation of flapping wings. In: Proceedings of the ASME IDETC/CIE 2013, August 4–7 Portland, Oregon, USA, DETC2013-12198 (2013)
- Ambrosio, J., Pombo, J., Rauter, F., Pereira, M.: A memory based communication in the co-simulation of multibody and finite element codes for pantograph–catenary interaction simulation. In: Bottasso, C.L. (ed.) *Multibody Dynamics: Computational Methods and Applications*, pp. 231–252. Springer, Berlin (2009)
- Ambrosio, J., Pombo, J., Pereira, M., Antunes, P., Mosca, A.: A computational procedure for the dynamic analysis of the catenary–pantograph interaction in high-speed trains. *J. Theor. Appl. Mech.* **50**(3), 681–699 (2012)
- Arnold, M., Clauss, C., Schierz, T.: Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation in V2.0. *Arch. Mech. Eng.* **60**(1), 75–94 (2013)
- Arnold, M.: Stability of sequential modular time integration methods for coupled multibody system models. *J. Comput. Nonlinear Dyn.* **5**, 1–9 (2010)
- Busch, M.: Continuous approximation techniques for co-simulation methods: analysis of numerical stability and local error. *Z. Angew. Math. Mech.* **96**(9), 1061–1081 (2016). <https://doi.org/10.1002/zamm.201500196>
- Datar, M., Stanciulescu, I., Negruț, D.: A co-simulation environment for high-fidelity virtual prototyping of vehicle systems. *Int. J. Veh. Syst. Model. Test.* **7**, 54–72 (2012)
- D'Silva, S., Sundaram, P., Ambrosio, J.: Co-Simulation Platform for Diagnostic Development of a Controlled Chassis System. SAE Technical Paper 2006-01-1058 (2006). <https://doi.org/10.4271/2006-01-1058>
- Eberhard, P., Gaugele, T., Heisel, U., Storchak, M.: A discrete element material model used in a co-simulated charpy impact test and for heat transfer. In: Proceedings 1st Int. Conference on Process Machine Interactions, 3–4 September, 2008, Hannover, Germany (2008)
- Gear, C.W., Wells, D.R.: Multirate linear multistep methods. *BIT Numer. Math.* **24**, 484–502 (1984)
- Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: State of the art. [arXiv:1702.00686](https://arxiv.org/abs/1702.00686)
- Gonzalez, F., Gonzalez, M., Cuadrado, J.: Weak coupling of multibody dynamics and block diagram simulation tools. In: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2009, August 30–September 2, 2009, San Diego, California, USA (2009)
- Gonzalez, F., Naya, M.A., Luaces, A., Gonzalez, M.: On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody Syst. Dyn.* **25**(4), 461–483 (2011)
- Gonzalez, F., Gonzalez, M., Mikkola, A.: Efficient coupling of multibody software with numerical computing environments and block diagram simulators. *Multibody Syst. Dyn.* **24**(3), 237–253 (2010). <https://doi.org/10.1007/s11044-010-9199-6>

15. Gu, B., Asada, H.H.: Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models. *J. Dyn. Syst. Meas. Control* **126**, 1–13 (2004). <https://doi.org/10.1115/1.1648307>
16. Haug, E.J.: Computer-Aided Kinematics and Dynamics of Mechanical Systems. Allyn & Bacon, Needham Heights (1989)
17. Kübler, R., Schiehlen, W.: Two methods of simulator coupling. *Math. Comput. Model. Dyn. Syst.* **6**, 93–113 (2000)
18. Lacourrière, C.: A parallel block iterative method for interactive contacting rigid multibody simulations on multicore PCs. In: Applied Parallel Computing. State of the Art in Scientific Computing, pp. 956–965 (2007)
19. Lacourrière, C., Nordfeldth, F., Linde, M.: A partitioning method for parallelization of large systems in realtime. In: Proceedings of the 3rd Joint International Conference on Multibody System Dynamics and the 7th Asian Conference on Multibody Dynamics, IMSD 2014, ACMD 2014, June 30–July 3, 2014 Bexco, Busan, Korea (2014)
20. Lehnart, A., Fleissner, F., Eberhard, P.: Using SPH in a co-simulation approach to simulate sloshing in tank vehicles. In: Proceedings SPHERIC4, 27–29 May, 2009, Nantes, France (2009)
21. Liao, Y.G., Du, H.I.: Co-simulation of multi-body-based vehicle dynamics and an electric power steering control system. *Proc. Inst. Mech. Eng., Proc., Part K, J. Multi-Body Dyn.* **215**, 141–151 (2001)
22. Malczyk, P., Fraczek, J.: Evaluation of parallel efficiency in modeling of mechanisms using commercial multibody solvers. *Arch. Mech. Eng. LVI*(3), 237–249 (2009)
23. Maple User Manual: <https://www.maplesoft.com>
24. Mraz, L., Valasek, M.: Parallelization of multibody system dynamics by additional dynamics. In: Proceedings of the ECCOMAS Thematic Conference Multibody Dynamics, 1–4 July, 2013, Zagreb, Croatia (2013)
25. Naya, M., Cuadrado, J., Dopico, D., Lugris, U.: An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and co-integration approaches. *Arch. Mech. Eng.* **LVIII**, 223–243 (2011)
26. Negruț, N., Melanz, D., Mazhar, H., Lamb, D., Jayakumar, P.: Investigating through simulation the mobility of light tracked vehicles operating on discrete granular terrain. *SAE Int. J. Passeng. Cars, Mech. Syst.* **6**, 369–381 (2013). <https://doi.org/10.4271/2013-01-1191>
27. Negruț, D., Tasora, A., Mazhar, H., Heyn, T., Hahn, P.: Leveraging parallel computing in multibody dynamics. *Multibody Syst. Dyn.* **27**, 95–117 (2012). <https://doi.org/10.1007/s11044-011-9262-y>
28. Negruț, D., Serban, R., Mazhar, H., Heyn, T.: Parallel computing in multibody system dynamics: why, when and how. *J. Comput. Nonlinear Dyn.* **9**(4), 041007 (2014). <https://doi.org/10.1115/1.4027313>
29. Pombo, J., Ambrosio, J.: Multiple pantograph interaction with catenaries in high-speed trains. *J. Comput. Nonlinear Dyn.* **7**(4), 041008 (2012)
30. Quaranta, G., Masarati, P., Mantegazza, P.: Multibody analysis of controlled aeroelastic systems on parallel computers. *Multibody Syst. Dyn.* **8**(1), 71–102 (2002). <https://doi.org/10.1023/A:1015894729968>
31. Schneider, F., Burger, M., Arnold, M., Simeon, B.: A new approach for force-displacement co-simulation using kinematic coupling constraints. *Z. Angew. Math. Mech.* (2017). <https://doi.org/10.1002/zamm.201500129>
32. Schweizer, B., Lu, D.: Predictor/corrector co-simulation approaches for solver coupling with algebraic constraints. *Z. Angew. Math. Mech.* (2014). <https://doi.org/10.1002/zamm.201300191>
33. Schweizer, B., Lu, D.: Semi-implicit co-simulation approach for solver coupling. *Arch. Appl. Mech.* (2014). <https://doi.org/10.1007/s00419-014-0883-5>
34. Schweizer, B., Lu, D.: Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. *Multibody Syst. Dyn.* (2014). <https://doi.org/10.1007/s11044-014-9422-y>
35. Schweizer, B., Li, P., Lu, D.: Explicit and implicit co-simulation methods: stability and convergence analysis for different solver coupling approaches. *J. Comput. Nonlinear Dyn.* (2014). <https://doi.org/10.1115/1.4028503>
36. Schweizer, B., Li, P., Lu, D.: Implicit co-simulation methods: stability and convergence analysis for solver coupling with algebraic constraints. *Z. Angew. Math. Mech.* (2015). <https://doi.org/10.1002/zamm.201400087>
37. Serban, R., Melanz, D., Li, A., Stanciulescu, I., Jayakumar, P., Negruț, D.: A GPU-based preconditioned Newton–Krylov solver for flexible multibody dynamics. *Int. J. Numer. Methods Eng.* **102**(9), 1585–1604 (2015)
38. Spreng, F., Eberhard, P., Fleissner, F.: An approach for the coupled simulation of machining processes using multibody system and smoothed particle hydrodynamics algorithms. *Theor. Appl. Mech. Lett.* **3**(1), 8–013005 (2013)
39. Solcia, T., Masarati, P.: Efficient multirate simulation of complex multibody systems based on free software. In: Proceedings of the ASME IDETC/CIE 2011, DETC2011-47306, August 28–31, 2011, Washington, DC, USA (2011)

40. Tomulik, P., Fraczek, J.: Simulation of multibody systems with the use of coupling techniques: a case study. *Multibody Syst. Dyn.* **25**(2), 145–165 (2011)
41. Tseng, F., Hulbert, G.: Network-distributed multibody dynamics simulation-gluing algorithm. In: Ambrósio, J., Schiehlen, W. (eds.) *Advances in Computational Multibody Dynamics*, IDMEC/IST, Lisbon, Portugal, pp. 521–540 (1999)
42. Valasek, M., Mraz, L.: Massive Parallelization of Multibody System Simulation. *Acta Polytech.* **52**(6) (2012)
43. Wang, J., Ma, Z.D., Hulbert, G.: A gluing algorithm for distributed simulation of multibody systems. *Nonlinear Dyn.* **34**, 159–188 (2003)
44. Wuensche, S., Clauß, C., Schwarz, P., Winkler, F.: Electro-thermal circuit simulation using simulator coupling. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **5**, 277–282 (1997)
45. Zierath, J., Woernle, C.: Contact modelling in multibody systems by means of a boundary element co-simulation and a Dirichlet-to-Neumann algorithm. In: Oñate, E. (ed.) *Computational Methods in Applied Sciences*. Springer, Berlin (2012)