THE ARCHIVE OF MECHANICAL ENGINEERING

VOL. LX 2013 Number 1

10.2478/meceng-2013-0005

Key words: multidisciplinary simulation, co-simulation, functional mock-up interface, FMI

MARTIN ARNOLD [*], *CHRISTOPH CLAUSS* [**], *TOM SCHIERZ* [*]

ERROR ANALYSIS AND ERROR ESTIMATES FOR CO-SIMULATION IN FMI FOR MODEL EXCHANGE AND CO-SIMULATION V2.0

Complex multi-disciplinary models in system dynamics are typically composed of subsystems. This modular structure of the model reflects the modular structure of complex engineering systems. In industrial applications, the individual subsystems are often modelled separately in different mono-disciplinary simulation tools. The *Functional Mock-Up Interface* (FMI) provides an interface standard for coupling physical models from different domains and addresses problems like export and import of model components in industrial simulation tools (*FMI for Model Exchange*) and the standardization of co-simulation interfaces in nonlinear system dynamics (*FMI for Co-Simulation*), see [10].

The renewed interest in algorithmic and numerical aspects of co-simulation inspired some new investigations on error estimation and stabilization techniques in FMI for Model Exchange and Co-Simulation v2.0 compatible co-simulation environments. In the present paper, we focus on reliable error estimation for communication step size control in this framework.

1. Introduction

Co-Simulation is a rather general approach to the simulation of coupled technical systems and coupled physical phenomena in engineering with focus on time-dependent problems. Theoretical and practical aspects of co-simulation were recently studied in the ITEA2 project MODELISAR (2008-2011). A *Functional Mock-Up Interface* (FMI) was developed that provides

[*] *Martin Luther University Halle-Wittenberg, NWF II – Institute of Mathematics, D - 06099 Halle (Saale), Germany; E-mail: martin.arnold@mathematik.uni-halle.de; Tom Schierz, SIMPACK AG, Friedrichshafener Strasse 1, D-82205 Gilching, Germany; E-mail: Tom.Schierz@simpack.de*

[**] *Fraunhofer Institute for Integrated Circuits IIS, Design Automation Division EAS, Zeunerstr. 38, D - 01069 Dresden, Germany; E-mail: christoph.clauß@eas.iis.fraunhofer.de*

an interface standard for coupling physical models from different domains and addresses problems like export and import of model components in industrial simulation tools (*FMI for Model Exchange*) and the standardization of co-simulation interfaces in nonlinear system dynamics (*FMI for Co-Simulation*), see [10].

Today, the interface standard is maintained and further developed within the Modelica Association Project "FMI". In August 2012, the fourth β-version of FMI for Model Exchange and Co-Simulation v2.0 was released [15] that supports advanced numerical techniques in co-simulation. Well known industrial simulation tools for applied dynamics support Version 1.0 of this standard and plan to support the forthcoming Version 2.0 in the near future, see the "Tools" tab of website [10] for up-to-date information.

FMI for Co-Simulation is an interface standard for the solution of time-dependent coupled systems consisting of subsystems that are continuous in time (model components that are described by time-dependent differential equations) or time-discrete (e.g., discrete controllers). In a block representation of the coupled system, the time continuous subsystems are represented by blocks with (internal) state variables $\boldsymbol{x}_j(t)$ that are connected to other subsystems (blocks) of the coupled problem by subsystem inputs $\boldsymbol{u}_j(t)$ and subsystem outputs $\boldsymbol{y}_j(t)$, see Fig. 1 and Section 2 below. In this framework, the physical connections between subsystems are represented by mathematical coupling conditions between the inputs $\boldsymbol{u}(t)$ and the outputs $\boldsymbol{y}(t)$ of all subsystems [14], [15].

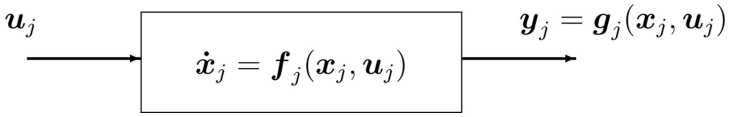

Fig. 1. Block-oriented representation of slave FMU's [15]

Co-Simulation exploits the modular structure of coupled problems in all stages of the simulation process with separate model setup, pre- and postprocessing for the individual subsystems in different simulation tools. During time integration, the simulation is again performed independently for all subsystems restricting the data exchange between subsystems to discrete communication points T_n, see [2]. In different contexts, the communication points T_n, the communication steps $T_n \to T_{n+1}$ and the communication step sizes $H_n := T_{n+1} - T_n$ are also known as sampling points (synchronization points), macro steps and sampling rates, respectively [15].

From the mathematical viewpoint, the coupling of different numerical solvers in a co-simulation environment results in a *modular* time integration method that substitutes the (unknown) subsystem inputs $\boldsymbol{u}_j(t)$ between the communication points T_n by some suitable approximation like polynomial

extrapolation or interpolation [2]. This *signal extrapolation* introduces additional error terms in the modular time integration and may furthermore result in numerical instability. FMI for Model Exchange and Co-Simulation v2.0 addresses these problems by interface routines supporting higher-order extrapolation and interpolation of subsystem inputs, communication step size control including step rejection and Jacobian-based linearly implicit stabilization techniques.

The present paper focuses on communication step size control and error estimation. For more information about Jacobian-based stabilization techniques we refer to [1] and the more recent results in [3], [18]. The paper is organized as follows: In Section 2, we give a sufficient criterion to exclude algebraic loops in a coupled system with block structure. Basic steps of a convergence analysis in this framework are presented in Section 3. Generalizing Richardson extrapolation techniques from ODE and DAE theory to modular time integration, we get reliable estimates for the local error. Theoretical and practical aspects are discussed in Section 4. The results of the theoretical analysis are verified by numerical tests for a quarter car model (Section 5). The paper ends with some conclusions in Section 6.

The numerical tests for the quarter car benchmark problem were performed in a Matlab based test environment. Recently, these results were reproduced using an FMI for Co-Simulation v1.0 compatible master that was developed at Fraunhofer IIS/EAS within the MODELISAR project [7], [19].

2. Block representation of coupled systems

Following the approach of Kübler and Schiehlen [14], FMI for Model Exchange and Co-Simulation v2.0 is based on a block representation of coupled systems. The mathematical analysis of modular time integration methods for coupled systems of ordinary differential equations (ODEs) [1] may be extended to this more general problem class if there are no algebraic loops in the system. The specific numerical problems that may result from a coupling by constraints or other algebraic loops in the system found much interest in the early days of co-simulation algorithms in system dynamics, see, e.g., [4], [14], but have today only limited practical relevance in industrial applications.

Therefore, we will start the theoretical investigations by a structural analysis of coupled systems in block representation to exclude systems with algebraic loops. Let subsystem "j" be described by its state and output equations

$$\left.\begin{aligned} \dot{\boldsymbol{x}}_j(t) &= \boldsymbol{f}_j(\boldsymbol{x}_j(t), \boldsymbol{u}_j(t), \boldsymbol{u}_{\mathrm{ex}}(t)) \\ \boldsymbol{y}_j(t) &= \boldsymbol{g}_j(\boldsymbol{x}_j(t), \boldsymbol{u}_j(t)) \end{aligned}\right\} \tag{1a}$$

with $\boldsymbol{x}_j$, $\boldsymbol{u}_j$, $\boldsymbol{y}_j$ denoting the state, input and output vectors and some external input $\boldsymbol{u}_{\mathrm{ex}}(t)$. The $r \geq 2$ subsystems are coupled by input-output relations

$$\boldsymbol{u}_j(t) = \boldsymbol{c}_j(\boldsymbol{y}_1(t), \ldots, \boldsymbol{y}_{j-1}(t), \boldsymbol{y}_{j+1}(t), \ldots, \boldsymbol{y}_r(t)), \quad (j = 1, \ldots, r). \tag{1b}$$

Summarizing all components $\boldsymbol{x}_j$, $\boldsymbol{y}_j$, $\boldsymbol{u}_j$, $\boldsymbol{f}_j$, $\boldsymbol{g}_j$, $\boldsymbol{c}_j$ in vector form, we get the coupled system in the more compact form

$$\left.\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{u}_{\mathrm{ex}}(t)) \\ \boldsymbol{y}(t) &= \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{u}(t) = \boldsymbol{c}(\boldsymbol{y}(t)) \end{aligned}\right\} \tag{2}$$

with $\boldsymbol{x}(t) := (\boldsymbol{x}_1^\top(t), \boldsymbol{x}_2^\top(t), \ldots, \boldsymbol{x}_r^\top(t))^\top, \ldots$. Eqs. (2) form a differential-algebraic equation (DAE) in variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{u}$ that may, however, be reduced straightforwardly to an ordinary differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}_{\mathrm{ex}}(t)) := \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}(t))), \boldsymbol{u}_{\mathrm{ex}}(t)) \tag{3a}$$

and separate output equations

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t)), \quad \boldsymbol{u}(t) = \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}(t))) \tag{3b}$$

whenever there is no direct feed-through in the subsystems, i.e., whenever

$$\boldsymbol{y}_j(t) = \boldsymbol{g}_j(\boldsymbol{x}_j(t)), \quad \frac{\partial \boldsymbol{g}_j}{\partial \boldsymbol{u}_j}(\boldsymbol{x}_j, \boldsymbol{u}_j) \equiv \boldsymbol{0}, \quad (j = 1, \ldots, r). \tag{4}$$

From the mathematical viewpoint, the ODE like structure (3) would be much more favourable than the general DAE system (2) but condition (4) is too restrictive and excludes, e.g., force displacement couplings between two mechanical systems.

Example 1 (Busch and Schweizer [6]) *Consider two 1-DOF oscillators with position and velocity coordinates (p_1, v_1) and (p_2, v_2), respectively, that are coupled by a linear spring with stiffness and damping parameters c and d resulting in a spring force*

$$F_{\mathrm{c}} = c(p_2 - p_1) + d(v_2 - v_1).$$

Variables (p_j, v_j) are components of state vectors $\boldsymbol{x}_j$, $(j = 1, 2)$. A displacement-displacement coupling *is realized by input-output couplings*

$$\boldsymbol{u}_1 = (u_1^1, u_1^2)^\top = \boldsymbol{y}_2 \in \mathbb{R}^2, \quad \boldsymbol{u}_2 = (u_2^1, u_2^2)^\top = \boldsymbol{y}_1 \in \mathbb{R}^2$$

and subsystem outputs $\boldsymbol{y}_j = (p_j, v_j)^\top$, *(*$j = 1, 2$*), such that there is no direct feed-through and condition (4) is satisfied. In both subsystems "j", the coupling spring force is evaluated as* $F_\mathrm{c} = (-1)^j(c(p_j - u_j^1) + d(v_j - u_j^2))$. *The same physical system may also be described by a* force-displacement coupling *with input-output couplings*

$$\boldsymbol{u}_1 = (u_1^1) = \boldsymbol{y}_2 \in \mathbb{R}\,, \quad \boldsymbol{u}_2 = (u_2^1, u_2^2)^\top = \boldsymbol{y}_1 \in \mathbb{R}^2$$

and subsystem outputs

$$\boldsymbol{y}_1 = \boldsymbol{g}_1(\boldsymbol{x}_1) = (p_1, v_1)^\top\,, \quad \boldsymbol{y}_2 = \boldsymbol{g}_2(\boldsymbol{x}_2, \boldsymbol{u}_2) = F_\mathrm{c} = c(p_2 - u_2^1) + d(v_2 - u_2^2)$$

resulting in a direct feed-through in the second subsystem such that (4) is violated for $j = 2$. *In the first subsystem, the coupling spring force is directly given by* $\boldsymbol{u}_1 = \boldsymbol{y}_2 = F_\mathrm{c}$. *In the second subsystem it is again evaluated as* $F_\mathrm{c} = c(p_2 - u_2^1) + d(v_2 - u_2^2)$.

In Example 1, the force-displacement coupling results in a slightly more complex coupling structure, but the overall coupled system is (as before) free of algebraic loops. For coupled systems of the general form (Eq. 3a) with direct feed-through in one or more subsystems, a refined structural analysis is necessary to exclude algebraic loops. For this analysis, the output equations in (Eq. 3a) are formally eliminated resulting in a system of n_u (coupled) nonlinear equations

$$\boldsymbol{u} = \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})) \tag{5}$$

in the n_u (scalar) components u^ν, $(\nu = 1, \ldots, n_u)$, of

$$\boldsymbol{u} = (\,\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_r^\top)^\top = (\,u^1, \ldots, u^{n_u}\,)^\top \in \mathbb{R}^{n_u}\,.$$

In (Eq. 5), component u^ν does not depend directly on component u^μ if $\partial c^\nu(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}))/\partial u^\mu \equiv 0$. In a structural sense, component u^ν may depend directly on component u^μ if

$$\left.\frac{\partial c^\nu(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}))}{\partial u^\mu}\right|_{\boldsymbol{x}=\boldsymbol{x}^*, \boldsymbol{u}=\boldsymbol{u}^*} \neq 0 \quad \text{for some argument } (\boldsymbol{x}^*, \boldsymbol{u}^*). \tag{6}$$

In that case, there is a *structural feed-through path of length* 1 from u^μ to u^ν. In Example 1, there are no structural feed-through paths in the case of displacement-displacement coupling. For force-displacement coupling, structural feed-through paths of length 1 point from u_2^1 to u_1^1 and from u_2^2 to u_1^1.

A *structural feed-through path of length* L from u^μ to u^ν consists of L connected structural feed-through paths of length 1. It is characterized by indices $\mu_l \in \{\,0, 1, \ldots, n_u\,\}$, $(\,l = 0, 1, \ldots, L\,)$, with $\mu_k \neq \mu_l$ if $0 < |k - l| < L$,

$(k, l = 0, 1, \ldots, L)$, $\mu_0 = \mu$, $\mu_L = \nu$ and structural feed-through paths of length 1 from $u^{\mu_{l-1}}$ to u^{μ_l}, $(l = 1, \ldots, L)$. The structural feed-through path from u^{μ} to u^{ν} is called *open* if $\mu \neq \nu$ and *closed* otherwise. In (Eq. 5), there are no closed structural feed-through paths of length $L = 1$ since subsystem input vector $\boldsymbol{u}_j$ does not depend directly on subsystem output vector $\boldsymbol{y}_j$, see (Eq. 1b). A trivial closed structural feed-through path of length $L = 2$ would result from output equations $\boldsymbol{y}_1 = \boldsymbol{u}_1$, $\boldsymbol{y}_2 = \boldsymbol{u}_2$ with coupling conditions $\boldsymbol{u}_1 = \boldsymbol{y}_2$, $\boldsymbol{u}_2 = \boldsymbol{y}_1$ (algebraic loop). In Example 1, there are no structural feed-through paths of length $L \geq 2$ and therefore also no closed structural feed-through paths.

Structural feed-through paths in (Eq. 5) may be studied conveniently by a directed graph G with n_u vertices "ν" that correspond to the components u^{ν}, $(\nu = 1, \ldots, n_u)$, of $\boldsymbol{u}$ in (Eq. 5), see, e.g., [[8], Appendix B.4] for a compact introduction to basics of graph theory. In this directed graph, an edge points from vertex "ν" to vertex "μ" whenever condition (Eq. 6) is satisfied. Each structural feed-through path of length L in (Eq. 5) corresponds to a path of length L in the associated graph G. On the other hand, a cycle in G represents a closed structural feed-through path in the system of nonlinear equations (Eq. 5).

To exclude algebraic loops in the coupled system (Eq.3), we suppose in the following that the directed graph G associated with (Eq. 5) is acyclic, i.e., free of cycles. This condition implies that the adjacency matrix $\mathbf{A}(G) \in \mathbb{R}^{n_u \times n_u}$ is nilpotent and there is some $M \in \mathbb{N}$, $M \leq n_u$, such that $(\mathbf{A}(G))^M = \mathbf{0}$, see [[8], Chapter 22]. Here, the n_u^2 elements $a_{\nu\mu}(G)$ of $\mathbf{A}(G)$ are given by $a_{\nu\mu} = 1$ if there is an edge from vertex "ν" to vertex "μ" and $a_{\nu\mu} = 0$ otherwise. Condition (Eq. 6) shows that the adjacency matrix represents the "structural" sparsity pattern of the Jacobian $\partial(\boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})))/\partial \boldsymbol{u}$. As a practical consequence, the nilpotency of $\mathbf{A}(G)$ implies

$$\prod_{l=1}^{M} \frac{\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}))}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}=\mathbf{x}^{[l]},\, \boldsymbol{u}=\mathbf{u}^{[l]}} \cdot \mathbf{Z}^{[l]} \equiv \mathbf{0} \tag{7}$$

for *any* system configurations $(\mathbf{x}^{[1]}, \mathbf{u}^{[1]})$, $\ldots$, $(\mathbf{x}^{[M]}, \mathbf{u}^{[M]})$ and any diagonal matrices $\mathbf{Z}^{[1]}$, $\ldots$, $\mathbf{Z}^{[M]} \in \mathbb{R}^{n_u \times n_u}$.

Example 2 *For systems without direct feed-through, graph G has no edges at all, see (Eq. 4) and (Eq. 6). Therefore, the displacement-displacement coupling in Example 1 results in the graph G in the left plot of Fig. 2 with* $\mathbf{A}(G) = \mathbf{0}_{4\times4}$. *For the force-displacement coupling, we obtain the graph G in*

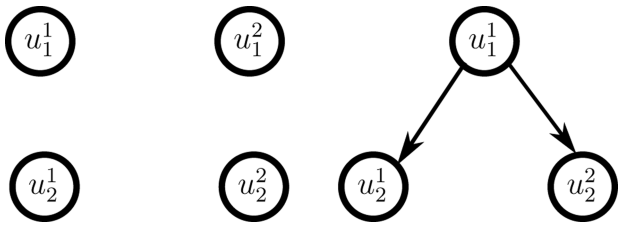

Fig. 2. Graphs representing the feed-through structure of the model in Example 1. Left plot: displacement-displacement coupling, Right plot: force-displacement coupling

the right plot of Fig. 2 and the non-zero but nilpotent adjacency matrix

$$\mathbf{A}(G) = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

with $(\mathbf{A}(G))^2 = \mathbf{0}$. *Since output equations and coupling equations are linear, the Jacobian* $\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}))/\partial \boldsymbol{u}$ *is constant and we get*

$$\frac{\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}))}{\partial \boldsymbol{u}}\,\mathbf{Z}^{[1]}\,\frac{\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}))}{\partial \boldsymbol{u}}\,\mathbf{Z}^{[2]} = \begin{pmatrix} 0 & -c & -d \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{Z}^{[1]}\begin{pmatrix} 0 & -c & -d \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{Z}^{[2]} = \mathbf{0}_{3\times 3} \tag{8}$$

for any diagonal matrices $\mathbf{Z}^{[1]}, \mathbf{Z}^{[2]} \in \mathbb{R}^{3\times 3}$.

3. Convergence analysis

Standard step size control algorithms in ODE and DAE time integration adjust the step size in each time step to guarantee that (an estimate) of the *local* error does not exceed some user-defined error bounds. A perturbation analysis shows that this strategy will bound the *global* error after a finite time interval of length T as well since local errors are amplified in time integration by a factor that is bounded by $\exp(L_0 T)$ with L_0 denoting the Lipschitz constant of the right hand side (in the ODE case). For coupled systems in block representation (Eq. 1), the situation is more complicated since order reduction phenomena have been reported for the local error analysis [5, Appendix 2.A] that do, however, not deteriorate the practically observed order of the global error.

In the present section, we discuss basic steps of a convergence analysis for modular time integration methods applied to coupled systems in block representation (Eq. 1) that provides a theoretical justification of this practically observed error behaviour. Local and global errors are not as directly

connected as in the classical ODE and DAE case but the communication step size control may nevertheless be based on local error estimates.

For the theoretical analysis, we follow the framework of [4] and neglect the discretization errors of the time integration in all subsystems to focus on the additional error terms that are introduced by coupling these subsystems in a co-simulation framework. In each communication step $T_n \to T_{n+1} = T_n + H$, the input vectors $\boldsymbol{u}_j(t)$ are approximated by some interpolation polynomial $\boldsymbol{\Psi}_j(t; \boldsymbol{u}_j(T_{n-k}), \ldots, \boldsymbol{u}_j(T_n), \boldsymbol{u}_j(T_{n+1}))$ of degree $\le k$, see [9]. An important special case is the signal extrapolation based on interpolation polynomials

$$\boldsymbol{\Psi}_j(t) = \sum_{\iota=0}^{k} \boldsymbol{u}_j(T_{n-\iota}) \prod_{\substack{l=0 \\ l \neq \iota}}^{k} \frac{t - T_{n-l}}{T_{n-\iota} - T_{n-l}} = \boldsymbol{u}_j(t) + O(H^{k+1}) \,. \tag{9}$$

Example 3 *In a brute force approach, the input vectors $\boldsymbol{u}_j(t)$ are simply frozen in each communication step $T_n \to T_{n+1} = T_n + H$ resulting in $k = 0$ and a constant extrapolation polynomial $\boldsymbol{\Psi}_j(t) \equiv \boldsymbol{u}_j(T_n)$, ($t \in [T_n, T_{n+1}]$). For higher accuracy, linear ($k = 1$) and quadratic ($k = 2$) interpolation is more favourable:*

$$\begin{aligned}
\boldsymbol{\Psi}_j(t) &= \boldsymbol{u}_j(T_n) + \frac{\boldsymbol{u}_j(T_n) - \boldsymbol{u}_j(T_{n-1})}{T_n - T_{n-1}}(t - T_n) \,, \quad (k = 1) \,, \\
\boldsymbol{\Psi}_j(t) &= \boldsymbol{u}_j(T_n) + \frac{\boldsymbol{u}_j(T_n) - \boldsymbol{u}_j(T_{n-1})}{T_n - T_{n-1}}(t - T_n) + \boldsymbol{\beta}_{j,n}(t - T_n)(t - T_{n-1}), \quad (k = 2),
\end{aligned}$$

with

$$\boldsymbol{\beta}_{j,n} := \Big(\frac{\boldsymbol{u}_j(T_n) - \boldsymbol{u}_j(T_{n-1})}{T_n - T_{n-1}} - \frac{\boldsymbol{u}_j(T_{n-1}) - \boldsymbol{u}_j(T_{n-2})}{T_{n-1} - T_{n-2}} \Big) / (T_n - T_{n-2}) \,.$$

The local error in communication step $T_n \to T_{n+1} = T_n + H$ compares for $t \in [T_n, T_{n+1}]$ the solution of (Eq. 2) with the solution of

$$\left.\begin{aligned}
\dot{\hat{\boldsymbol{x}}}(t) &= \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \boldsymbol{\Psi}(t), \boldsymbol{u}_{\mathrm{ex}}(t)) \\
\hat{\boldsymbol{y}}(t) &= \boldsymbol{g}(\hat{\boldsymbol{x}}(t), \boldsymbol{\Psi}(t)) \\
\hat{\boldsymbol{u}}(t) &= \boldsymbol{c}(\hat{\boldsymbol{y}}(t))
\end{aligned}\right\}, \quad (t \in [T_n, T_{n+1}]) \,, \quad \hat{\boldsymbol{x}}(T_n) = \boldsymbol{x}(T_n) \,. \tag{10}$$

The perturbation analysis for ODE initial value problems [20] shows

$$\begin{aligned}
\|\hat{\boldsymbol{x}}(T_{n+1}) - \boldsymbol{x}(T_{n+1})\| &\le C_x \big(\exp(L_0(T_{n+1} - T_n)) - 1 \big) \max_{t \in [T_n, T_{n+1}]} \|\boldsymbol{\Psi}(t) - \boldsymbol{u}(t)\| \\
&= O(H \cdot H^{k+1}) = O(H^{k+2}) \,, \\
\|\hat{\boldsymbol{u}}(T_{n+1}) - \boldsymbol{u}(T_{n+1})\| &\le C_u \|\hat{\boldsymbol{y}}(T_{n+1}) - \boldsymbol{y}(T_{n+1})\| = O(H^{k+1}) \,,
\end{aligned}$$

see (Eq. 9). Following the convergence analysis for linear multistep methods in the DAE case [11], we get a coupled error recursion for the global errors $\boldsymbol{\epsilon}_n^{\boldsymbol{x}}$, $\boldsymbol{\epsilon}_n^{\boldsymbol{u}}$ in differential and algebraic solution components:

$$\|\boldsymbol{\epsilon}_{n+1}^{\boldsymbol{x}}\| \leq (1 + O(H))\|\boldsymbol{\epsilon}_n^{\boldsymbol{x}}\| + O(H) \sum_{\iota=0}^{k} \|\boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}}\| + O(H^{k+2}), \tag{11a}$$

$$\boldsymbol{\epsilon}_{n+1}^{\boldsymbol{u}} = \sum_{\iota=0}^{k} \mathbf{J}_n \mathbf{Z}_{n-\iota} \boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}} + O(1)\|\boldsymbol{\epsilon}_n^{\boldsymbol{x}}\| + O(H) \sum_{\iota=0}^{k} \|\boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}}\| + O(H^{k+1}) \tag{11b}$$

with

$$\mathbf{J}_n := \frac{\partial \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}))}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}=\boldsymbol{x}(T_n),\, \boldsymbol{u}=\boldsymbol{u}(T_n)}, \quad \mathbf{Z}_{n-\iota} := \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{u}_{n-\iota}}(t; \boldsymbol{u}_{n-k}, \ldots, \boldsymbol{u}_n, \boldsymbol{u}_{n+1})\Big|_{t=T_{n+1}}.$$

The linear interpolation polynomial $\boldsymbol{\Psi}_j(t)$ in Example 3 results, e.g., in $k = 1$ and $\mathbf{Z}_n = 2\mathbf{I}$, $\mathbf{Z}_{n-1} = -\mathbf{I}$ since $\boldsymbol{\Psi}_j(T_{n+1}) = 2\boldsymbol{u}_j(T_n) - \boldsymbol{u}_j(T_{n-1})$. For constant extrapolation ($k = 0$) with $\boldsymbol{\Psi}_j(T_{n+1}) = \boldsymbol{u}_j(T_n)$, Jacobian $\mathbf{Z}_n$ is given by the identity matrix $\mathbf{I}$.

Estimate (Eq. 11b) provides an error bound for $\|\boldsymbol{\epsilon}_{n+1}^{\boldsymbol{u}}\|$ in terms of $O(1)\sum_\iota \|\boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}}\|$ that does in general not guarantee zero-stability and convergence of the modular time integration method [14]. There is a risk of exponential instability unless an additional contractivity condition is satisfied [4].

However, the $\sum_\iota \mathbf{J}_n \mathbf{Z}_{n-\iota} \boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}}$ term at the right hand side of (Eq. 11b) may be eliminated by repeated application of this error estimate if condition (Eq. 7) is satisfied for some finite $M > 0$. Substituting index n in (Eq. 11b) by $n - \iota - 1$, we get

$$\boldsymbol{\epsilon}_{n-\iota}^{\boldsymbol{u}} = \sum_{\kappa=0}^{k} \mathbf{J}_{n-\iota-1} \mathbf{Z}_{n-\iota-1-\kappa} \boldsymbol{\epsilon}_{n-\iota-1-\kappa}^{\boldsymbol{u}} + \\ + O(1)\|\boldsymbol{\epsilon}_{n-\iota-1}^{\boldsymbol{x}}\| + O(H) \sum_{\kappa=0}^{k} \|\boldsymbol{\epsilon}_{n-\iota-1-\kappa}^{\boldsymbol{u}}\| + O(H^{k+1}) \tag{11c}$$

for all $\iota = 0, 1, \ldots, k$. This error recursion is inserted in the right hand side of (Eq. 11b) resulting in

$$\boldsymbol{\epsilon}_{n+1}^{\boldsymbol{u}} = \sum_{\iota=0}^{k} \sum_{\kappa=0}^{k} \mathbf{J}_n \mathbf{Z}_{n-\iota} \mathbf{J}_{n-\iota-1} \mathbf{Z}_{n-\iota-1-\kappa} \boldsymbol{\epsilon}_{n-\iota-1-\kappa}^{\boldsymbol{u}} + \\ + O(1) \sum_{l=0}^{2k+1} \left(\|\boldsymbol{\epsilon}_{n-l}^{\boldsymbol{x}}\| + H\|\boldsymbol{\epsilon}_{n-l}^{\boldsymbol{u}}\|\right) + O(H^{k+1}). \tag{11d}$$

For the coupled 1-DOF oscillators in Example 1, we have $\mathbf{J}_n = \mathbf{0}_{4\times4}$ in the case of displacement-displacement coupling and $\mathbf{J}_n \neq \mathbf{0}_{3\times3}$ but $\mathbf{J}_n\mathbf{Z}_{n-\iota}\mathbf{J}_{n-\iota-1}\mathbf{Z}_{n-\iota-1-\kappa} = \mathbf{0}_{3\times3}$ in the case of force-displacement coupling, see (Eq. 8). For both settings, estimate (Eq. 11d) simplifies to

$$\|\boldsymbol{\epsilon}^{\boldsymbol{u}}_{n+1}\| \leq \overline{C}_u \sum_{l=0}^{M(k+1)-1} \left(\|\boldsymbol{\epsilon}^{\boldsymbol{x}}_{n-l}\| + H\|\boldsymbol{\epsilon}^{\boldsymbol{u}}_{n-l}\|\right) + O(H^{k+1}) \tag{11e}$$

with $M = 2$. In the general case, estimate (Eq. 11e) is obtained repeating the transformation steps from (Eq. 11b) to (Eq. 11d) another $M - 2$ times and exploiting the identity (Eq. 7) that follows from the basic assumption that the coupled system (3) is represented by an acyclic directed graph G, see Section 2.

From the coupled error recursion (Eq. 11a,e), we get global errors

$$\|\boldsymbol{\epsilon}^{\boldsymbol{x}}_n\| = O(H^{k+1}), \quad \|\boldsymbol{\epsilon}^{\boldsymbol{u}}_n\| = O(H^{k+1}),$$

i.e., convergence with order $k + 1$ on finite time intervals, see the classical convergence analysis for DAE time integration methods in [11], its application to modular time integration in [4] and the recent extension of these results to multistep signal extrapolation in [3].

This convergence result gives a theoretical justification for communication step size control algorithms being based on (estimates for) the local error since the global error on finite time intervals may be bounded in terms of the local error if the coupled system (Eq. 3) is free of algebraic loops and condition (Eq. 7) is satisfied for some finite $M > 0$. It shows furthermore that order reduction in the local error will typically not affect the global error if there are no closed structural feed-through paths in the coupled system since by definition the length of all open structural feed-through paths is bounded by the dimension n_u of (5) such that $M \leq n_u$ in (Eq. 7).

4. Local error estimates

For communication step size control, an appropriate estimate of the local error is compared to user-defined error bounds (tolerances), see [4]. Richardson extrapolation is a time-consuming but reliable way to estimate local errors in ODE and DAE time integration. Substantial savings of computing time result from an algorithmic modification that is tailored to the FMI co-simulation framework. In the present section both approaches are studied by an asymptotic error analysis. Numerical test results for a quarter car benchmark problem are presented in Section 5 below.

To keep the notation compact, we restrict the theoretical analysis of the error estimates to pure polynomial signal extrapolation in all subsystems that allows us to perform and to analyse a communication step $T_n \to T_{n+1} = T_n + H$ for all subsystems in parallel. As in (Eq. 9), we consider a polynomial $\overline{\boldsymbol{u}}(t)$ that interpolates the analytical solution $\boldsymbol{u}(t)$ of (Eq. 2) at the $k+1$ equidistant previous communication points $T_{n-\iota}$, $(\iota = 0, 1, \ldots, k)$. Classical estimates for the error of polynomial interpolation [9] prove

$$\boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(t) := \overline{\boldsymbol{u}}(t) - \boldsymbol{u}(t) = -\frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} \prod_{\iota=0}^{k} (t - T_{n-\iota}) + O(H^{k+2}) \tag{12}$$

for all $t \in [T_n, T_{n+2}]$. Replacing input function $\boldsymbol{u}(t)$ in (Eq. 2) by $\overline{\boldsymbol{u}}(t)$, we get approximate solutions $\overline{\boldsymbol{x}}(t)$, $\overline{\boldsymbol{y}}(t)$ with $\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(T_n) = \mathbf{0}$ and

$$\left.\begin{aligned} \dot{\boldsymbol{\Delta}}_{\overline{\boldsymbol{x}}}(t) &= \mathbf{A}_n\, \boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(t) + \mathbf{B}_n\, \boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(t) + O(H^{k+2}) \\ \boldsymbol{\Delta}_{\overline{\boldsymbol{y}}}(t) &= \mathbf{C}_n\, \boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(t) + \mathbf{D}_n\, \boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(t) + O(H^{k+2}) \end{aligned}\right\} \tag{13}$$

for

$$\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(t) := \overline{\boldsymbol{x}}(t) - \boldsymbol{x}(t)\,, \quad \boldsymbol{\Delta}_{\overline{\boldsymbol{y}}}(t) := \overline{\boldsymbol{y}}(t) - \boldsymbol{y}(t)\,, \quad (\,t \in [T_n, T_{n+2}]\,)\,.$$

In (Eq. 13), system matrices $\mathbf{A}_n$, $\mathbf{B}_n$, $\mathbf{C}_n$, $\mathbf{D}_n$ denote the Jacobians $\boldsymbol{f}_{\boldsymbol{x}}$, $\boldsymbol{f}_{\boldsymbol{u}}$, $\boldsymbol{g}_{\boldsymbol{x}}$, $\boldsymbol{g}_{\boldsymbol{u}}$ evaluated at $\boldsymbol{x} = \boldsymbol{x}(T_n)$, $\boldsymbol{u} = \boldsymbol{u}(T_n)$. The leading error term in $\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}$ is obtained as solution of a linear time invariant system resulting in

$$\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(T_n + aH) = -\mathbf{B}_n \int_0^a \prod_{\iota=0}^{k} (\iota + s)\,\mathrm{d}s \cdot \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+2} + O(H^{k+3})\,, \tag{14}$$

see also (Eq. 12). In the next communication step $T_{n+1} \to T_{n+2} = T_n + 2H$, input function $\boldsymbol{u}(t)$ is substituted by a polynomial $\overline{\overline{\boldsymbol{u}}}(t)$ that interpolates $\boldsymbol{c}(\overline{\boldsymbol{y}}(t)) = \boldsymbol{c}(\boldsymbol{g}(\overline{\boldsymbol{x}}(t), \overline{\boldsymbol{u}}(t)))$ at $t = T_{n+1}$ and $\overline{\boldsymbol{u}}(t) = \boldsymbol{u}(t) = \boldsymbol{c}(\boldsymbol{y}(t))$ at $t = T_{n+1-\iota}$, $(\iota = 1, \ldots, k)$. Eq. (5) implies

$$\boldsymbol{u}(T_{n+1}) = \boldsymbol{c}\big(\boldsymbol{g}(\boldsymbol{x}(T_{n+1}), \boldsymbol{u}(T_{n+1}))\big)$$

and

$$\begin{aligned} \overline{\overline{\boldsymbol{u}}}(T_{n+1}) &= \overline{\boldsymbol{u}}(T_{n+1}) - \boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(T_{n+1}) + \boldsymbol{c}\big(\boldsymbol{g}(\overline{\boldsymbol{x}}(T_{n+1}), \overline{\boldsymbol{u}}(T_{n+1}))\big) - \boldsymbol{c}\big(\boldsymbol{g}(\boldsymbol{x}(T_{n+1}), \boldsymbol{u}(T_{n+1}))\big) \\ &= \overline{\boldsymbol{u}}(T_{n+1}) - (\mathbf{I} - \mathbf{L}_n \mathbf{D}_n) \boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(T_{n+1}) + O(H^{k+2}) \end{aligned}$$

with $\mathbf{L}_n := (\partial \boldsymbol{c}/\partial \boldsymbol{y})(\boldsymbol{y}(T_n))$ since $\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(T_{n+1}) = O(H^{k+2})$, see (Eq. 14). According to (Eq. 12), polynomial $\overline{\boldsymbol{u}}(t)$ approximates $\boldsymbol{u}(t)$ up to $O(H^{k+1})$ and we get

$$\boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(T_{n+1}) = -\boldsymbol{u}^{(k+1)}(T_n) \cdot H^{k+1} + O(H^{k+2})\,.$$

The Lagrange form (Eq. 9) of the interpolation polynomial shows

$$\begin{aligned}\overline{\overline{\boldsymbol{u}}}(t) - \overline{\boldsymbol{u}}(t) &= (\overline{\overline{\boldsymbol{u}}}(T_{n+1}) - \overline{\boldsymbol{u}}(T_{n+1})) \prod_{l=1}^{k} \frac{t - T_{n+1-l}}{T_{n+1} - T_{n+1-l}} \\ &= H(\mathbf{I} - \mathbf{L}_n \mathbf{D}_n) \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} (k+1) \prod_{l=1}^{k} (t - T_{n+1-l}) + O(H^{k+2}). \end{aligned} \tag{15}$$

Since estimate (Eq. 15) has the same basic structure as estimate (Eq. 12), the *local error* at $t = T_{n+2}$

$$\boldsymbol{l}^{\boldsymbol{x}}_{n+2} := \underline{\underline{\boldsymbol{\Delta}}}_{\boldsymbol{x}}(T_{n+2}) = \boldsymbol{\Delta}_{\overline{\overline{\boldsymbol{x}}}}(T_{n+2}) + \int_{T_{n+1}}^{T_{n+2}} (\boldsymbol{f}(\overline{\overline{\boldsymbol{x}}}(t), \overline{\overline{\boldsymbol{u}}}(t)) - \boldsymbol{f}(\overline{\boldsymbol{x}}(t), \overline{\boldsymbol{u}}(t))) \, \mathrm{d}t$$

may be expressed as

$$\boldsymbol{l}^{\boldsymbol{x}}_{n+2} = -\mathbf{B}_n (c_k \mathbf{I} + d_k \mathbf{L}_n \mathbf{D}_n) \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+2} + O(H^{k+3}) \tag{16}$$

with constants

$$d_k := (k+1) \int_1^2 \prod_{\iota=1}^{k} (\iota + s - 1) \, \mathrm{d}s \,, \quad c_k = \int_0^2 \prod_{\iota=0}^{k} (\iota + s) \, \mathrm{d}s - d_k = 2 \int_0^1 \prod_{\iota=0}^{k} (\iota + s) \, \mathrm{d}s \,. \tag{17}$$

For coupled systems with direct feed-through in some of the subsystems, we have $\mathbf{D}_n \neq \mathbf{0}$ and the local error $\boldsymbol{l}^{\boldsymbol{y}}_{n+2}$ in the output variables $\boldsymbol{y}$ is dominated by the error in components $\boldsymbol{u}$ at $t = T_{n+2}$:

$$\begin{aligned}\boldsymbol{l}^{\boldsymbol{u}}_{n+2} := \underline{\underline{\boldsymbol{\Delta}}}_{\boldsymbol{u}}(T_{n+2}) &= \overline{\overline{\boldsymbol{u}}}(T_{n+2}) - \boldsymbol{u}(T_{n+2}) = \overline{\overline{\boldsymbol{u}}}(T_{n+2}) - \overline{\boldsymbol{u}}(T_{n+2}) + \overline{\boldsymbol{u}}(T_{n+2}) - \boldsymbol{u}(T_{n+2}) \\ &= -(\mathbf{I} + (k+1)\mathbf{L}_n \mathbf{D}_n) \boldsymbol{u}^{(k+1)}(T_n) \cdot H^{k+1} + O(H^{k+2}) \,, \end{aligned} \tag{18}$$

see (Eq. 12) and (Eq. 15). Summarizing (Eq. 16) and (Eq. 18), we end up with

$$\begin{aligned}\boldsymbol{l}^{\boldsymbol{y}}_{n+2} &= \underline{\underline{\boldsymbol{\Delta}}}_{\boldsymbol{y}}(T_{n+2}) = \mathbf{D}_n \underline{\underline{\boldsymbol{\Delta}}}_{\boldsymbol{u}}(T_{n+2}) + \mathbf{C}_n \underline{\underline{\boldsymbol{\Delta}}}_{\boldsymbol{x}}(T_{n+2}) + O(c_{\mathbf{D}} H^{k+2} + H^{k+3}) \\ &= -\big((k+1)! \, \mathbf{D}_n (\mathbf{I} + (k+1)\mathbf{L}_n \mathbf{D}_n) + \\ &\quad + H \mathbf{C}_n \mathbf{B}_n (c_k \mathbf{I} + d_k \mathbf{L}_n \mathbf{D}_n)\big) \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+1} + O(c_{\mathbf{D}} H^{k+2} + H^{k+3}) \,. \end{aligned} \tag{19}$$

Constant $c_{\mathbf{D}}$ is set to $c_{\mathbf{D}} = 0$ if $\partial \boldsymbol{g} / \partial \boldsymbol{u} \equiv \mathbf{0}$ and to $c_{\mathbf{D}} = 1$ otherwise. For systems without direct feed-through, see (Eq. 4), we have $\mathbf{D}_n = \mathbf{0}$ and $c_{\mathbf{D}} = 0$ and the right hand side in (Eq. 19) simplifies substantially:

$$\boldsymbol{l}^{\boldsymbol{y}}_{n+2} = -c_k \mathbf{C}_n \mathbf{B}_n \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+2} + O(H^{k+3}) \,. \tag{20}$$

For error estimation by Richardson extrapolation, we consider in each time interval $[T_n, T_{n+2}]$ a second numerical solution $\tilde{\boldsymbol{x}}(t)$ with $\tilde{\boldsymbol{x}}(T_n) = \boldsymbol{x}(T_n)$ and an input function $\tilde{\boldsymbol{u}}(t)$ that is defined by the interpolation polynomial for data points $(T_{n-2\iota}, \boldsymbol{u}(T_{n-2\iota}))$ with $T_{n-2\iota} = T_n - 2\iota H$, $(\iota = 0, 1, \ldots, k)$. As before, the interpolation error $\boldsymbol{\Delta}_{\tilde{\boldsymbol{u}}}(t)$ introduces errors $\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}(t)$ and $\boldsymbol{\Delta}_{\tilde{\boldsymbol{y}}}(t)$ in state and output variables that may be studied by a perturbation analysis. We get

$$\boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}(T_{n+2}) = -\tilde{c}_k \mathbf{B}_n \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+2} + O(H^{k+3}) \tag{21}$$

with

$$\tilde{c}_k := \int_0^2 \prod_{\iota=0}^{k} (2\iota + s)\,\mathrm{d}s = 2\int_0^1 \prod_{\iota=0}^{k} (2\iota + 2\tilde{s})\,\mathrm{d}\tilde{s} = 2^{k+1} c_k\,,$$

see (Eq. 17). Similar to estimate (Eq. 12), the interpolation error in components $\boldsymbol{u}$ at $t = T_{n+2}$ is given by

$$\begin{aligned} \boldsymbol{\Delta}_{\tilde{\boldsymbol{u}}}(T_{n+2}) &= -\frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} \prod_{\iota=0}^{k} (T_{n+2} - T_{n-2\iota}) + O(H^{k+2}) \\ &= -2^{k+1} \boldsymbol{u}^{(k+1)}(T_n) H^{k+1} + O(H^{k+2})\,. \end{aligned} \tag{22}$$

Estimates (Eq. 21) and (Eq. 22) yield an error bound for the output variables:

$$\begin{aligned} \boldsymbol{\Delta}_{\tilde{\boldsymbol{y}}}(T_{n+2}) &= \mathbf{D}_n \boldsymbol{\Delta}_{\tilde{\boldsymbol{u}}}(T_{n+2}) + \mathbf{C}_n \boldsymbol{\Delta}_{\tilde{\boldsymbol{x}}}(T_{n+2}) + O(c_{\mathbf{D}} H^{k+2} + H^{k+3}) \\ &= -2^{k+1}\big((k+1)!\,\mathbf{D}_n + H c_k \mathbf{C}_n \mathbf{B}_n\big) \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+1} + O(c_{\mathbf{D}} H^{k+2} + H^{k+3}). \end{aligned} \tag{23}$$

In ODE and DAE time integration, the comparison of the numerical results for a double-step with (small) step size H and a single (large) step with step size $2H$ allows us to estimate precisely the leading error term of the local error [11]. For modular time integration, this error estimate is given by [13]

$$\mathbf{EST}_{\mathrm{Rich}} := \frac{\tilde{\boldsymbol{y}}(T_{n+2}) - \bar{\bar{\boldsymbol{y}}}(T_{n+2})}{2^{k+1} - 1} = \frac{\boldsymbol{\Delta}_{\tilde{\boldsymbol{y}}}(T_{n+2}) - \boldsymbol{\Delta}_{\bar{\bar{\boldsymbol{y}}}}(T_{n+2})}{2^{k+1} - 1}\,. \tag{24}$$

The comparison of (Eq. 19), (Eq. 23) and (Eq. 24) shows

$$\mathbf{EST}_{\mathrm{Rich}} = \boldsymbol{l}^{\boldsymbol{y}}_{n+2} + \frac{2^{k+1}(k+1)}{2^{k+1} - 1} \mathbf{D}_n \mathbf{L}_n \mathbf{D}_n \boldsymbol{u}^{(k+1)}(T_n) H^{k+1} + O(c_{\mathbf{D}} H^{k+2} + H^{k+3})\,. \tag{25}$$

There is a qualitative difference to classical ODE and DAE theory: In the context of co-simulation, Richardson extrapolation may give asymptotically

wrong results if $\mathbf{L}_n\mathbf{D}_n \neq \mathbf{0}$, i.e., for coupled systems with direct feed-through in at least one subsystem. If there are no subsystems with direct feed-through and condition (Eq. 4) is satisfied, $\mathbf{EST}_{\mathrm{Rich}}$ reproduces, however, all components of the local error in the output variables correctly up to higher order terms. Therefore, this error estimate is considered to be one of the most reliable ways to approximate local errors in modular time integration.

In the ITEA2 project MODELISAR, several modifications of error estimate (Eq. 24) were tested [16] to reduce the large extra effort for computing $\tilde{\boldsymbol{y}}(t)$. Neglecting all details of practical implementation, the intermediate results $\overline{\boldsymbol{x}}(T_{n+1})$ and $\tilde{\boldsymbol{x}}(T_{n+1})$ coincide for constant extrapolation ($k = 0$) of input function $\boldsymbol{u}(t)$ since $\overline{\boldsymbol{u}}(t) = \tilde{\boldsymbol{u}}(t) = \boldsymbol{u}(T_n)$, $(t \in [T_n, T_{n+1}])$, in that case. From the view point of numerical efficiency, it would be favourable to use also for higher-order extrapolation ($k \geq 1$) one and the same approximation $\overline{\boldsymbol{u}}(t)$ of the input function $\boldsymbol{u}(t)$ for both numerical solutions in the first communication step $T_n \rightarrow T_{n+1} = T_n + H$ and to restrict the use of different input functions to the second communication step, i.e., to $t \in [T_{n+1}, T_{n+2}]$.

In that way, co-simulation may proceed with a large communication step $T_n \rightarrow T_{n+2} = T_n + 2H$ of size $2H$ that is temporarily interrupted at $t = T_{n+1}$ to provide input data $\overline{\boldsymbol{y}}(T_{n+1})$ and $\boldsymbol{c}(\overline{\boldsymbol{y}}(T_{n+1}))$ for the second numerical solution to be used for error estimation. Alternatively, a small communication step $T_n \rightarrow T_{n+1} = T_n + H$ may be completed in the classical way and the two different numerical solutions on time interval $[T_{n+1}, T_{n+2}]$ are evaluated in parallel. With this second strategy, no subsystem solver has to go backward in time and the practical implementation might be simplified. Note that both strategies are analytically equivalent but give slightly different results in a practical implementation since the internal (micro) step size sequences in the subsystems depend on the length of the communication steps.

With the notations of this section, the modified error estimate is given by

$$\mathbf{EST}_{\mathrm{mod}} := \frac{\overline{\boldsymbol{y}}(T_{n+2}) - \overline{\overline{\boldsymbol{y}}}(T_{n+2})}{c_{k,\mathrm{mod}} - 1} = \frac{\boldsymbol{\Delta}_{\overline{\boldsymbol{y}}}(T_{n+2}) - \boldsymbol{\Delta}_{\overline{\overline{\boldsymbol{y}}}}(T_{n+2})}{c_{k,\mathrm{mod}} - 1} \tag{26}$$

with some suitable constant $c_{k,\mathrm{mod}}$. To adjust this constant, we observe

$$\boldsymbol{\Delta}_{\overline{\boldsymbol{x}}}(T_{n+2}) = -\overline{c}_k \mathbf{B}_n \frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!} H^{k+2} + O(H^{k+3}) \quad \text{with} \quad \overline{c}_k := \int_0^2 \prod_{\iota=0}^{k} (\iota + s)\,\mathrm{d}s \tag{27}$$

and set $c_{k,\mathrm{mod}} := \overline{c}_k / c_k$ to get an asymptotically correct error estimate for systems without direct feed-through, see (Eq. 20). For constant extrapolation, we have $c_{0,\mathrm{mod}} = 2 = 2^{k+1}$ since $\overline{c}_0 = \tilde{c}_0$. For linear and quadratic extrapolation, $c_{1,\mathrm{mod}} = 14/5$ and $c_{2,\mathrm{mod}} = 32/9$ are used.

For coupled systems with direct feed-through in one of the subsystems, we consider the error in input function $\overline{\boldsymbol{u}}(t)$ at $t = T_{n+2}$:

$$\begin{aligned}\boldsymbol{\Delta}_{\overline{\boldsymbol{u}}}(T_{n+2}) &= -\frac{\boldsymbol{u}^{(k+1)}(T_n)}{(k+1)!}\prod_{\iota=0}^{k}(T_{n+2}-T_{n-\iota}) + O(H^{k+2}) \\ &= -(k+2)\boldsymbol{u}^{(k+1)}(T_n)H^{k+1} + O(H^{k+2})\,.\end{aligned}$$

This error is $k+2$ times larger than the first error term in $\boldsymbol{l}^{\boldsymbol{u}}_{n+2}$, see (Eq. 18), implying the necessary condition $c_{k,\mathrm{mod}} = k+2$ for an asymptotically correct modified error estimate (Eq. 26) in the case of direct feed-through in one of the subsystems ($\mathbf{D}_n \neq \mathbf{0}$). This condition is trivially satisfied for constant extrapolation but slightly violated in the linear and quadratic case ($k = 1$ and $k = 2$).

5. Numerical test: Benchmark Quarter car

Fig. 3 shows the strongly simplified model of a quarter car with two point masses m_{w} and m_{c} representing the primary mass (unsprung mass) and the secondary mass (spring mounted mass), respectively, see [17]. The point masses have a vertical degree of freedom. They are coupled by a linear spring-damper element F_{susp} representing the secondary suspension. Tire forces are considered by an additional spring-damper element F_{tire} between unsprung mass and ground.

The road profile is considered as external system input $z(t)$. At $t = 0$ the system starts in its equilibrium state $x_{\mathrm{c}} = x_{\mathrm{w}} = 0$ and is immediately excited by the jump discontinuity of z. For model parameters according to Knorr [12], Fig. 4 shows the vertical displacements illustrating the different time scales in unsprung and spring mounted mass, respectively.

In co-simulation, both subsystems are integrated separately by standard ODE integrators with very fine error tolerances. The subsystems are either coupled by a displacement-displacement coupling or by a force-displacement coupling, see Example 1. For various values of the communication step size H, the local error $\|\boldsymbol{l}^{\boldsymbol{y}}_{n+2}\|_2$ of two consecutive communication steps $T_n \to T_n + H$, $T_n + H \to T_n + 2H$, see (Eq. 19), is compared to the error estimates $\mathbf{EST}_{\mathrm{mod}}$ and $\mathbf{EST}_{\mathrm{Rich}}$. Note that the absolute values of $\boldsymbol{l}^{\boldsymbol{y}}_{n+2}$ are substantially larger in the case of force-displacement coupling (Fig. 6) since $\boldsymbol{u}_1 = \boldsymbol{y}_2 = F_{\mathrm{susp}}$ in that case, see Example 1. For displacement-displacement coupling (Fig. 5), the components of $\boldsymbol{y}$ are given by position and velocity coordinates x_{c}, $\dot{x}_{\mathrm{c}}$, x_{w}, $\dot{x}_{\mathrm{w}}$ that are of much smaller magnitude than $|F_{\mathrm{susp}}|$.

$$
\begin{aligned}
m_{\mathrm{c}}\ddot{x}_{\mathrm{c}} &= F_{\mathrm{susp}}(x_{\mathrm{c}}, \dot{x}_{\mathrm{c}}, x_{\mathrm{w}}, \dot{x}_{\mathrm{w}}) \\
m_{\mathrm{w}}\ddot{x}_{\mathrm{w}} &= F_{\mathrm{tire}}(x_{\mathrm{w}}, \dot{x}_{\mathrm{w}}, \boldsymbol{u}_{\mathrm{ex}}(t)) - F_{\mathrm{susp}}(x_{\mathrm{c}}, \dot{x}_{\mathrm{c}}, x_{\mathrm{w}}, \dot{x}_{\mathrm{w}}) \\
\boldsymbol{u}_{\mathrm{ex}}(t) &= (\, z(t),\, \dot{z}(t)\,)^{\top}, \quad z(t) = \begin{cases} 0 & \text{if } t \le 0 \\ 0.1 & \text{if } t > 0 \end{cases} \\
F_{\mathrm{susp}} &= k_{\mathrm{c}}(x_{\mathrm{w}} - x_{\mathrm{c}}) + d_{\mathrm{c}}(\dot{x}_{\mathrm{w}} - \dot{x}_{\mathrm{c}}) \\
F_{\mathrm{tire}} &= k_{\mathrm{w}}(z - x_{\mathrm{w}}) + d_{\mathrm{w}}(\dot{z} - \dot{x}_{\mathrm{w}})
\end{aligned}
$$

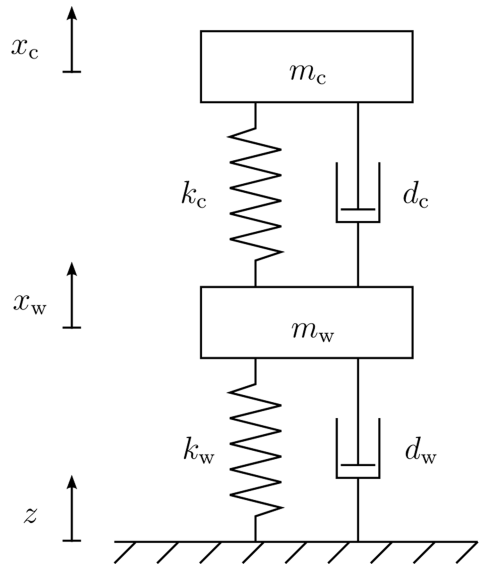

Fig. 3. Benchmark Quarter car: Equations of motion

Furthermore, the numerical test results in Figs. 5 and 6 illustrate $\|\boldsymbol{l}^{\boldsymbol{y}}_{n+2}\|_2 = O(H^{k+2})$ for systems without direct feed-through (displacement-displacement coupling, $c_{\mathbf{D}} = 0$, Fig. 5) and the reduced order $\|\boldsymbol{l}^{\boldsymbol{y}}_{n+2}\|_2 = O(H^{k+1})$ for systems with direct feed-through in at least one subsystem (force-displacement coupling, $c_{\mathbf{D}} = 1$, Fig. 6), see (Eq. 19) and (Eq. 20), respectively. In both cases, the new, alternative error estimate $\mathbf{EST}_{\mathrm{mod}}$ is as reliable as the classical estimate $\mathbf{EST}_{\mathrm{Rich}}$.

m_c	400.0 kg
m_w	40.0 kg
k_c	15000.0 N/m
k_w	150000.0 N/m
d_c	1000.0 Ns/m
d_w	0.0 Ns/m

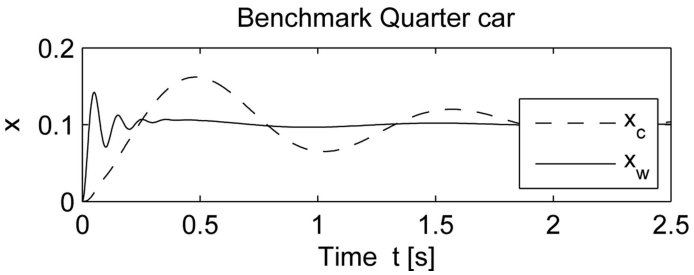

Fig. 4. Benchmark Quarter car: Model parameters and solution [12]

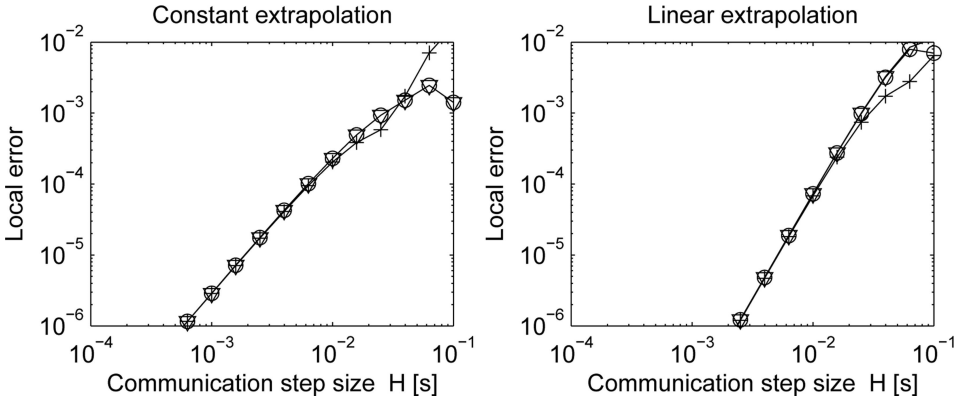

Fig. 5. Benchmark Quarter car, co-simulation with displacement-displacement coupling: Local error ("+") and error estimates $\mathbf{EST}_{\mathrm{Rich}}$, $\mathbf{EST}_{\mathrm{mod}}$ ("∘", "∇")

The local error analysis in Section 4 and numerical test results like the ones in Figs. 5 and 6 are important prerequisites for a theoretically justified communication step size control. The end-user, however, is not interested in *local* errors but in *global* ones. The convergence analysis in Section 3 shows that neither for displacement-displacement coupling nor for force-displacement coupling the global errors $\boldsymbol{\epsilon}_n^{\boldsymbol{x}}$, $\boldsymbol{\epsilon}_n^{\boldsymbol{y}}$, $\boldsymbol{\epsilon}_n^{\boldsymbol{u}}$ suffer from order reduction. Moreover, the global errors $\boldsymbol{\epsilon}_n^{\boldsymbol{x}}$ are expected to be in the same order of magnitude for both types of coupling since the state vector $\boldsymbol{x}$ has in both cases the same components x_c, $\dot{x}_c$, x_w and $\dot{x}_w$.

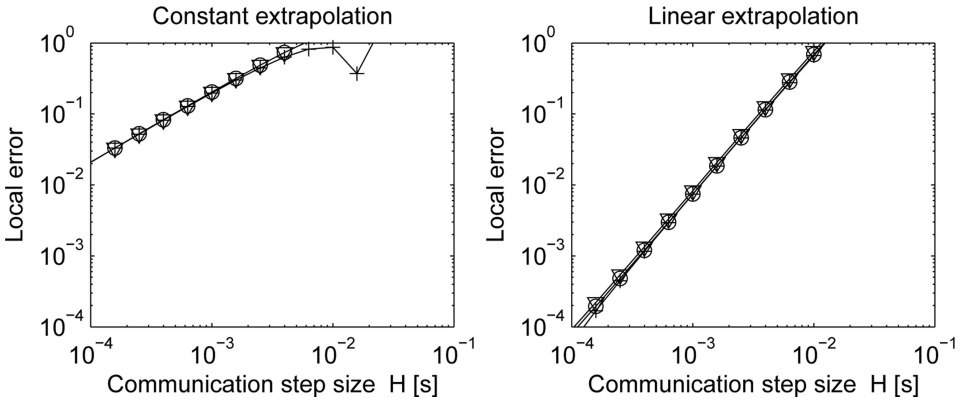

Fig. 6. Benchmark Quarter car, co-simulation with force-displacement coupling: Local error ("+") and error estimates $\mathbf{EST}_{\mathrm{Rich}}$, $\mathbf{EST}_{\mathrm{mod}}$ ("$\circ$", "∇")

Note that for a practical implementation of local error estimates like $\mathbf{EST}_{\mathrm{Rich}}$ and $\mathbf{EST}_{\mathrm{mod}}$ in communication step size control, the discretization errors of the (micro) time integration in the individual subsystems can no longer be neglected. This open problem is subject of future research, see also [19] for some preliminary results.

6. Conclusions

The numerical efficiency of co-simulation algorithms may be improved substantially by higher-order approximations of subsystem inputs and by variable communication step sizes. A strict mathematical analysis shows that the global error is bounded in terms of local errors if there are no algebraic loops in the coupled system. Local error estimates based on Richardson extrapolation techniques and some modifications have been studied in detail proving their efficiency in systems without direct feed-through like mechanical systems with displacement-displacement coupling. A force-displacement coupling results in direct feed-through in one of the subsystems and may deteriorate the favourable asymptotic properties of classical error estimation strategies.

Acknowledgement

The authors gratefully acknowledge the fruitful cooperation in the European ITEA2 project "MODELISAR - From system modelling to S/W running on the vehicle" (2008-2011) and valuable discussions with J. Bastian, T. Blochwitz (Dresden), H. Elmqvist, H. Olsson (Lund), M. Otter (Oberpfaffenhofen) and other partners of the MODELISAR consortium. The work was supported by the German Minister of Education and Research, BMBF

projects 01IS08002N and 03MS633A. The research on the convergence analysis for modular time integration methods applied to coupled systems in block representation was inspired by fruitful discussions with M. Busch (Kassel).

Manuscript received by Editorial Board, October 10, 2012;
final version, November 01, 2012.

REFERENCES

[1] M. Arnold: Multi-rate time integration for large scale multibody system models. In P. Eberhard, editor, *IUTAM Symposium on Multiscale Problems in Multibody System Contacts*, pages 1-10. Springer, 2007.

[2] M. Arnold: Numerical methods for simulation in applied dynamics. In M. Arnold and W. Schiehlen, editors, *Simulation Techniques for Applied Dynamics*, volume 507 of *CISM Courses and Lectures*, p. 191-246. Springer, Wien New York, 2009.

[3] M. Arnold: Stability of sequential modular time integration methods for coupled multibody system models. *J. Comput. Nonlinear Dynam.*, 5:031003, 2010.

[4] M. Arnold, M. Günther: Preconditioned dynamic iteration for coupled differential-algebraic systems. *BIT Numerical Mathematics*, 41:1-25, 2001.

[5] M. Busch: *Zur effizienten Kopplung von Simulationsprogrammen*. PhD thesis, Universität Kassel, Fachbereich Maschinenbau, 2012.

[6] M. Busch, B. Schweizer: Numerical stability and accuracy of different co-simulation techniques: Analytical investigations based on a 2-DOF test model. In *Proc. of The* 1st *Joint International Conference on Multibody System Dynamics, May 25-27, 2010*, Lappeenranta, Finland, 2010.

[7] C. Clauß, M. Arnold, T. Schierz, J. Bastian: Master zur Simulatorkopplung via FMI. In X. Liu-Henke, editor, *Tagungsband der ASIM/GI-Fachgruppen STS und GMMS, Wolfenbüttel, 23.02.-24.02.2012*, Wolfenbüttel, 2012. Ostfalia Hochschule für Angewandte Wissenschaften.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein: *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 2nd edition, 2001.

[9] P. Deuflhard, A. Hohmann: *Numerical Analysis in Modern Scientific Computing: An Introduction*. Number 43 in Texts in Applied Mathematics. Springer, 2nd edition, 2003.

[10] FMI: The Functional Mockup Interface. https://www.fmi-standard.org/.

[11] E. Hairer, G. Wanner: *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin Heidelberg New York, 2nd edition, 1996.

[12] S. Knorr: Multirate-Verfahren in der Co-Simulation gekoppelter dynamischer Systeme mit Anwendung in der Fahrzeugdynamik. Master Thesis, University Ulm, Faculty of Mathematics and Economics, 2002.

[13] R. Kübler: *Modulare Modellierung und Simulation mechatronischer Systeme*. Fortschritt-Berichte VDI Reihe 20, Nr. 327. VDI-Verlag GmbH, Düsseldorf, 2000.

[14] R. Kübler, W. Schiehlen: Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, 6:93-113, 2000.

[15] Modelica Association Project FMI: Functional Mockup Interface for Model Exchange and Co-Simulation v2.0 beta 4. https://www.fmi-standard.org/downloads, August 2012.

[16] H. Olsson: Private communication, June 2011.

[17] P. Popp, W.O. Schiehlen: *Ground Vehicle Dynamics*. Springer-Verlag, Berlin Heidelberg, 2010.

[18] T. Schierz, M. Arnold: Stabilized overlapping modular time integration of coupled differential-algebraic equations. *Applied Numerical Mathematics*, 62:1491-1502, 2012.
[19] T. Schierz, M. Arnold, C. Clauß: Co-Simulation with communication step size control in an FMI compatible master algorithm. In M. Otter and D. Zimmer, editors, *Proc. of the* 9th *International Modelica Conference, September 3-5, 2012*, Munich, Germany, 2012.
[20] W. Walter: *Ordinary Differential Equations*. Number 182 in Graduate Texts in Mathematics. Springer, 1998.

Analiza błędów i estymatory błędów wspólnej stymulacji w Funkcjonalnym Interfejsie Modelowania FMI wersji 2.0

Streszczenie

Złożone multidyscyplinarne modele stosowane w dynamice systemów są zwykle skonstruowane z podsystemów. Modularna struktura modelu odzwierciedla modularną strukturę złożonych systemów technicznych. W zastosowaniach przemysłowych poszczególne podsystemy są często modelowane indywidualnie przy pomocy różnych multidyscyplinarnych narzędzi symulacyjnych. Funkcjonalny interfejs modelowania (*Functional Mock-up Interface, FMI*) spełnia rolę standardowego interfejsu do łączenia modeli fizycznych z różnych dziedzin i pomaga rozwiązać problemy importu i eksportu elementów modelu w przemysłowych narzędziach modelowania (*FMI for Model Exchange*), lub standaryzacji interfejsów wspólnej stymulacji w dynamice systemów nieliniowych (*FMI for Co-Stimulation*), por. [10]. Odżywa na nowo zainteresowanie algorytmicznymi i numerycznymi aspektami wspólnej stymulacji, co zainspirowało do podjęcia wielu nowych badań nad estymacją błędów i technikami stabilizacji w interfejsie FMI wersji 2.0 w kompatybilnych środowiskach stymulacji. W prezentowanym artykule autorzy koncentrują się na wiarygodnej estymacji błędów przy sterowaniu rozmiarem kroku komunikacji w ramach tego interfejsu.