An Overview of the State of the Art in Co-Simulation and Related Methods

Irene Hafner[1*], Niki Popper[1,2]

[1]dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria; *irene.hafner@dwh.at

[2]TU Wien, Institute of Information Systems Engineering, Favoritenstraße 9-11, 1040 Vienna, Austria

SNE 31(4), 2021, 185-200, DOI: 10.11128/sne.31.on.10582
Received: 2021-09-10; Revised: 2021-11-03;
Accepted: 2021-11-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. In this paper, we present an overview of existing and ongoing developments in multirate and co-simulation. These are structured into thematical sections, providing general information on the state of the art while additionally allowing an impression of the progress in developments.

The first sections cover research on co-simulation of ODE and DAE systems, including different coupling methods along with comparisons and stability studies. This is followed by a description of standards for co-simulation, specific developments such as frameworks and investigations on hybrid co-simulation or partitioned methods. In conclusion, general strategies for the development and validation of coupling methods and summarized information on methods and challenges are given.

Introduction

One of the major challenges in research on multirate and co-simulation is the delineation of literature belonging to this area: the term “co-simulation” itself has surfaced shortly before the millennium and although it has become more commonly known since, some authors conduct co-simulation as understood in Definition 1, but do not use this precise term[1].

Definition 1 (Co-Simulation [1])**.** Co-simulation is the coupling of two or more simulations which differ in at least one of the following aspects:

- simulation tool
- solver algorithm
- step size.

While raising no claim for all-encompassing consideration, the following overview includes some publications without usage of the term co-simulation that nevertheless provides the basis of scientific research this area. Chronologically arranged details on the mentioned literature allowing a perspective on the “history of co-simulation" may be found in [2].

1 Beginnings in Classical Co-simulation: Coupling of ODEs

The first investigations on multirate and co-simulation have started on ODE systems, motivated by differing stiffness properties and time constants in system parts [3, 4, 5] or aiming at faster computation by parallelization [6]. The approaches vary between solutions with the same, yet adaptive step size [3], an adaptive approach with order control [5] and the introduction of waveform iteration [7]. Propositions regarding consistency depending on the used extrapolation order are found in [4, 5, 8]. What these investigations have in common is that the considered ODE IVP can be divided into two (or more) partial systems as depicted in (1):

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(t,\boldsymbol{x}_1,\boldsymbol{x}_2), \quad \boldsymbol{x}_1(t_0) = \boldsymbol{x}_{1,0} \tag{1a}$$

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(t,\boldsymbol{x}_1,\boldsymbol{x}_2), \quad \boldsymbol{x}_2(t_0) = \boldsymbol{x}_{2,0} \tag{1b}$$

2 Coupling Methods for DAEs

Owned in particular to applications in mechanical systems, research on co-simulation has soon extended to differential-algebraic equation systems. These can be

[1] Throughout this paper, the terminology from [1, 2] is applied. If needed, the reader is invited to look up the definition of specific terms (such as strong coupling, multirate simulation, hybrid simulation, etc.) there.

represented either as systems of ODEs which are coupled by algebraic constraints (see f.i. [9]), or systems of DAEs coupled by output-input dependencies (as in [10]). This means that the algebraic part can be restricted to the coupling equations (given for two subsystems in (2))

$$\begin{aligned}
&\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(t,\boldsymbol{x}_1,\boldsymbol{u}_1), \quad \boldsymbol{x}_1(t_0) = \boldsymbol{x}_{1,0}\\
&\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(t,\boldsymbol{x}_2,\boldsymbol{u}_2), \quad \boldsymbol{x}_2(t_0) = \boldsymbol{x}_{2,0}\\
&0 = \boldsymbol{g}(t,\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{u}_1,\boldsymbol{u}_2)\\
&\text{with time varying inputs } \boldsymbol{u}_i, i = 1,2
\end{aligned} \tag{2}$$

or be part of every subsystem, shown for two subsystems in (3).

$$\begin{aligned}
&\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(t,\boldsymbol{x}_1,\boldsymbol{x}_2), \quad \boldsymbol{x}_1(t_0) = \boldsymbol{x}_{1,0}\\
&\boldsymbol{y}_1 = \boldsymbol{g}_1(t,\boldsymbol{x}_1,\boldsymbol{u}_1)\\
&\boldsymbol{u}_1 = \boldsymbol{L}_1\boldsymbol{y}_2\\
&\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(t,\boldsymbol{x}_1,\boldsymbol{x}_2), \quad \boldsymbol{x}_2(t_0) = \boldsymbol{x}_{2,0}\\
&\boldsymbol{y}_2 = \boldsymbol{g}_2(t,\boldsymbol{x}_2,\boldsymbol{u}_2)\\
&\boldsymbol{u}_2 = \boldsymbol{L}_2\boldsymbol{y}_1\\
&\text{with the elements of } \boldsymbol{L}_i \text{ being equal to}\\
&\text{zero or one.}
\end{aligned} \tag{3}$$

Specific coupling methods range from methods to regularize high-index DAEs [9, 11, 12] to automatic algorithms for the calculation of calling sequence [13] and linking of models [14]. The following sections cover certain kinds of coupling methods, therein iterative approaches (Section 2.1), master algorithms with different choices of macro step size (Section 2.2) and methods specialized in the decomposition and coupling of mechanical systems (Section 2.3). This arrangement may not be seen as classification (which may be found in [1]) but simply as means to provide a better overview owned to the multitude of referenced publications. In Section 2.4, works comparing two or more coupling approaches with respect to stability, accuracy or performance are presented. Investigations on stability and error estimates for co-simulation are found in Section 2.5.

2.1 Iterative methods

Iterative coupling methods, *waveform relaxation* (WR) in particular seem to have been introduced by [15] for DAEs, while the first mention including convergence theorems for certain methods applied to ODEs is found in [7]. In general, WR starts with a Gauß-Seidl or Jacobi type master step that is then repeated until desired tolerances are met. Thereby, a contractivity condition has to be fulfilled to guarantee stability and convergence [15, 16].

Throughout the years, dynamic iteration occurs time and again in different variations and improvements: The iterative approach presented by [17] utilizes reduced order models, [16] and [18] introduce preconditioning to counter instabilities while [19] present an iterative algorithm showing similarities to the sliding mode control method (cf. [12]) and the algorithm of [20, 21] uses interface Jacobians for stabilization. [22] extends the application on PDAEs and discusses emerging stability issues. In [23], a possibility for step size control is presented in addition to convergence criteria for coupled DAE systems in general.

2.2 Choice of macro steps

In this section, methods employing a dynamic choice of the macro step are presented. Most of these are adaptive algorithms where the macro step size, at which all subsystem simulators communicate, is chosen according to varying estimates. [24] and [25] realize automatic adaption of macro step sizes via a predictor-corrector method while [26] takes into account eigen frequencies of the overall and/or partial systems instead of local error estimates. [27] include an iterative approach with *increasing* macro size which is reduced again if a maximum of iterations is reached.

[28], on the other hand, present an algorithm without common macro steps, where the subsystems are solved sequentially with their individual step size, determining after every step the slowest und thus next system to be executed. A similar approach without synchronized time steps is applied by [29].

[30] point out challenges in macro step size control such as slow-down by small step sizes and error calculations, accuracy loss in case of large steps or, specific to co-simulation, the unknown influences between macro and micro steps.

2.3 Decomposition and coupling of mechanical systems

Coupling techniques for mechanical systems described as DAEs are assembled in this section. Due to the specific structure of these systems' description, several investigations on their decomposition ([31, 32, 33, 34]) and further, gluing in the form of different force-force (also called "T-T"), force-displacement (T-X), and displacement-displacement (X-X) coupling approaches (cf. [1, 35]) are made. On the one hand, the corresponding literature can be distinguished depending on these gluing strategies: [36] present an X-X strategy, [37] and [38] a T-T method and [39] a T-X method, while the works of [35] compare all three and consider systems coupled by applied forces/torques ([40, 35]) and also systems coupled by reaction forces/torques [41, 42, 43]. In addition, they apply different stabilization techniques: by additional Lagrange multipliers ([40]), consideration of derivatives or integrals of coupling conditions ([44]), or Baumgarte stabilization ([43]). Iterative methods are found in [37, 38, 34]; semi-implicit (i.e. predictor-corrector) approaches are considered f.i. by [41, 25]. Furthermore, [39] apply automatic partitioning and parallel computing.

2.4 Comparisons

In this section, comparisons regarding performance, accuracy, stability or suitability among different co-simulation methods or versus a monolithic approach are presented. [45, 46, 47] present and compare different strong coupling schemes to simulate fluid-structure interaction. Comparisons of loose with strong coupling schemes for the application in building energy systems are performed in [48, 49], concluding that selecting one of these methods comes down to a choice between performance and independent time steps or accuracy. Regarding the possibility of modularity in multiphysics system simulation, [25] comes to the conclusion that classical co-simulation is advantageous compared to coupling of dynamic with static subsystems. [50] even implements a framework with the aim of comparing protocols for data exchange and different coupling methods.

Rather than algorithms themselves, different implementations of co-simulation masters are compared by [51]. [29] compare non-iterative slowest first and fastest first approaches with inter- and extrapolation polynomials of varying degrees with a – maybe for some researchers frustrating yet crucial – conclusion that the choice of the best coupling algorithm has to be exercised individually for every given problem.

2.5 Stability and error estimates

To quantify the worth of coupling methods, these have to be investigated for the numerical effects they have on separately nicely working integration algorithms.

Already in 1984, [5] use error estimates for the truncation error to adapt the macro step size. In general, the order of the global error of the coupled method is bounded by the error of the subsystem solvers and the extrapolation method [8, 52, 53, 54]. In general, consistency is maintained when consistent methods are co-simulated, but maybe of lower order [55].

Further error estimates, based on Richardson extrapolation, can be found in [56] and [57]; an investigation on relative consistency by calculating the defect in [58]. [59] quantify the convergence rate of co-simulation with more than two participating subsystems.

While in the area of partitioned methods, investigations on stability have been published since the 1980s (see Section 5), they gain currency only since the year 2000 for classical loose coupling schemes. As the field of numerics of differential equations and differential algebraic equations itself comes as a vast area of research, the combination of different methods out of this area is even harder to investigate from a general point of view. That generalized stability analysis is difficult to accomplish is also pointed out by [60]: "A detailed stability analysis for modular time-integration methods is technically very complicated since it has to take into account several types of stiff coupling terms and different extrapolation and interpolation methods". Hence, many studies on stability of coupling methods are done on systems with certain limitations, such as constant extrapolation [61]. To obtain higher accuracy, however, a higher extrapolation order may be preferred, which can increase stability issues. These can be met by methods for stabilization such as iteration [10, 16], asynchronous algorithms [28, 29] or weighting algorithms [62]. For the latter, detailed error estimates regarding extrapolations of different order are given in [57]. An interesting outcome is that the reliability of estimates can depend on the kind of DAE coupling (T-T, X-T, or X-X, see Section 2.3).

Some promising stabilization techniques, such as the bilinear delay line by [63] or the introduction of filters by [10], require alteration of the models themselves, which is often not possible with complex problems of the integrate-and-collaborate kind.

The approach found in [64] stands out as they aim to increase stability by energy conservation between co-simulated systems, thereby using power bonds to calculate energy residuals. Stabilization of strongly coupled systems is addressed by [65].

[10] deduce that zero-stability cannot be guaranteed for loose coupling co-simulation in case algebraic loops occur and [61] shows that for sequential algorithms, the order in which the subsystems are executed is crucial for the stability properties of their co-simulation. Even in case of general convergence, the sequence of subsystem execution can influence the *order* of convergence [66].

3 Standards for Co-simulation

The variety of co-simulation methods and tools to be coupled with their origin from different fields of application has led to the desire of unification, which is aimed by the specification of standards. Still, these are constantly revised by the developers and also extended by other researchers to meet specific requirements. The two most popular standards which are also frequently found in the literature, the *High Level Architecture* and the *Functional Mockup Interface*, are presented here along with the DEV&DESS formalism. The latter – whether it may or may not be regarded as standard for co-simulation (cf. Section 2.12 in [2]) – constitutes an important approach that therefore also occurs occasionally, be it directly utilized or adapted, in the literature presented in this chapter.

3.1 High Level Architecture

The High Level Architecture (HLA) has been specified by the US Department of Defense to address the need for reuse and interoperability of simulations within the department. It provides an architecture defining functional elements, interfaces and design rules for simulation applications and a common framework for the definition of specific system architectures [67]. The HLA is software and programming language independent.

Its key functional components are *federates*, the *runtime infrastructure* (RTI) and the *runtime interface*. Federates can range from computer simulations to manned simulators and even interfaces to live players: the representation of a federate is not restricted as long as it allows the interaction with other objects through data exchanges via services from the RTI. The RTI is a distributed operating system offering these services for interaction and federation management. The runtime interface specification defines a standardized manner of interaction between the federates and the RTI independently from the implementation. Monitoring of simulation activities and interfaces to live participants such as control systems are also supported.

Formally, the HLA is defined by the following three components: *object model template*, *interface specification*, and the *HLA rules*. Different timing services by the HLA are described in [68].

It is made clear in [67] that while the HLA provides the minimum essential tools for interoperability, it is itself insufficient to guarantee interoperability.

3.2 Functional Mockup Interface

The Functional Mockup Interface (FMI) is a standard for model exchange and co-simulation initiated by the project MODELISAR and now maintained and developed by the Modelica Association. In a nutshell, the FMI defines the manner in which Functional Mockup Units (FMUs) have to be built so they can be imported by tools serving as master orchestrator and the functionalities and interfaces for the latter. When an FMU *for model exchange* is exported, the tool where the respective model has been implemented translates it into a dynamic system model in C-code with inputs and outputs. The models can contain events as well as differential, algebraic or discrete equations. In the FMI *for co-simulation*, not only the model but also the solution algorithm is included in the exported code.

Master algorithms can then define points in time where participating FMUs exchange data and control this data exchange. In addition to the C-code file, an FMU contains an XML file with the definition of input and output variables and other model information. Further C-functions for the setup of co-simulation minions or execution of model equations and optional data such as icons or documentation are also included in the zip-file (extension ".fmu") which finally constitutes a complete FMU. In the current version of the standard (FMI 2.0, see [69, 70]), the interfaces for model exchange and co-simulation are unified. Additional features such as getting and setting an FMU state (thus potentially enabling rollback) are introduced, but not mandatory for tools that support the FMI. Input and output dependencies of variables and their derivatives (important for algebraic loop detection) or Jacobian information (potentially needed for implicit integration methods or linearization) can also be included in an FMU.

The great potential and renown but also drawbacks and possibilities for improvements are assessed in an empirical survey [71]. Some of the main difficulties are accounted for by the optional features of the FMI, many of which are not supported by most (in particular open source) tools that often do not even properly define which features they support and which they do not. This hampers the implementation of coupling methods requiring specific functionalities such as simulator rollback, information on derivatives, or input-output dependencies. Another problem regarding discrete event or hybrid co-simulation is the requirement of time passing between two synchronization references, which means that simultaneous events cannot be handled by several exchanges of data at the same time step. This has led to extensions to the FMI standard f.i. by [72], who propose an extension by a procedure returning an upper bound for the FMU's acceptable step size, thus allowing adaptive master steps without requiring rollback, or [73], who aims to encode different formalisms such as state machines, discrete event, and synchronous dataflow as FMUs. These extensions naturally are not supported by all tools currently supporting the FMI 2.0 itself[2].

Different, but similar formalizations for FMU execution are presented in [74, 72, 75]. In all these, the authors argue that the validity of a master algorithm depends on the input-output dependencies inside FMUs, an information often not available as it is not required in the FMU description according to the standard alone. [74] present an algorithm based on topological ordering of a graph constructed according to input/output dependencies and further information such as feed-through and reactivity.

3.3 DEVS-based formalisms

The Discrete Event System Specification (DEVS) is a formalism to describe hierarchically structured Discrete Event systems based on systems theory. Similarly, the Differential Equation System Specification (DESS) allows the description of ODE systems. Both have been introduced by [76] and combined for the description of hybrid systems to the DEV&DESS (Discrete Event System & Differential Equation System Specification) formalism. On the deepest level of hierarchy, an *atomic DEVS* can be described as a set of inputs, outputs, states, internal and external transition functions, an output function and a time advance function. Instead of transition functions and the time advance function, an *atomic DESS* contains a *rate of change function* corresponding to the right side of an ODE. In contrast to DESS of Moore type, where the output function has only states in its argument, for Mealy type DESS the output function may depend directly on the inputs as well. In an *atomic DEV&DESS*, both are combined, resulting in discrete and continuous inputs, outputs, states, transition and output functions, a rate of change function and, in addition, a state event condition function. Two or more DEVS (or DESS, DEV&DESS respectively) can be combined into a *coupled DEVS* (or DESS or DEV&DESS), enhancing clarity and supporting modularity. The problem of concurrent events can be tackled by parallel DEVS (P-DEVS), where concurrency is resolved locally in every DEVS. Hybrid P-DEVS (introduced by [77]) are designed to represent discrete and continuous systems as parallel DEV&DESS. Since the DEVS constitutes a formalism, it is software independent. Specific implementations are found in [78, 77, 79, 80].

4 Specific Applications and Developments

This section covers on the one hand specifically implemented frameworks for co-simulation (Section 4.1) and on the other hand developments for a particular model description (hybrid systems in Section 4.2, FEM in Section 4.3) or application (Section 4.4).

4.1 Frameworks

The introduction of frameworks has become more and more popular to allow easy "plug-and-play" co-simulation. However, many frameworks have again been designed motivated by a specific problem or area of application, such as building simulation [81], automotive systems [82], traffic [83], multi-domain physical [84] or cyber-physical systems [85] and are limited to the co-simulation of certain tools, leaving gaps aimed to be filled by further developments. What is more is that these seemingly simple "enablers" of co-simulation bear the risk that systems are not properly checked for stability properties but rashly coupled, which can be amended by notwithstanding mindful consideration and inspection of every user.

[2] A complete list of tools supporting the FMI 2.0 can be found on https://fmi-standard.org/tools/

Many recent, independent developments respect the FMI standard [86, 87, 88, 89]. [68] even utilizes the HLA as well as the FMI. An implementation of a framework extending the FMI to allow hybrid co-simulation is found in [90, 91], see also Section 4.2, where these can be found along further frameworks that are specifically tailored to support hybrid co-simulation.

Other frameworks implement multi-threading with FMUs by deployment on a cluster (as in [86]) or on multiple-core machines [87], by which supra-linear speedup can be achieved.

4.2 Hybrid (co-)simulation

Hybrid systems – in the sense of combined continuous time (CT) and discrete event (DE) systems – have been an ever-present challenge of special interest within modeling and simulation. Only recently, co-simulation has emerged as a possible solution approach that brings along advantages but also approach-specific complications. Although several investigations considered in this section are not focused on *co*-simulation, the peculiarities as well as methods for hybrid simulation frequently apply regardless whether the combination of DE with CT approaches is realized via co-simulation or integrated models.

This pertains for instance to [92], who presents an overview of phenomena in hybrid simulation reported in the literature: event handling, run-time equation processing, discontinuous state changes, event iteration, chattering, and comparing Dirac pulses. These, of course, are equally important issues in hybrid co-simulation. Solutions for event respectively zero crossing detection are addressed by [93] and [94], event ordering by [95], chattering avoidance by [96] and [93], zeno-behavior by [93], and debugging in hybrid simulations by [97].

[98] describes events update schemata and presents a generic methodology for developing hybrid co-simulation tools. For a similar purpose, formalisms have been introduced, f.i. by [99], who proposes the Heterogeneous Flow System Specification (HFSS), or [94], who formalize the FMI, taking input-output dependencies and abstraction of functions into regard to create a non-cyclic graph of the overall system. [79] present a DEVS wrapper for hybrid co-simulation of FMUs implemented in MECSYCO using the DEV&DESS standard.

The work of [75] shall be emphasized at this point, as they define a range of requirements for hybrid co-simulation standards along with test components and acceptance criteria.

In many specific approaches, one part is controlled respectively set back by the other: methods with the DE simulation as master are found in [100], CT simulation is taken as master by [101], and [98] employs both of these options. [102] present parallel approaches with potential rollback in both parts. [68] uses an iterative approach and [103] apply step size control. Comparisons of different hybrid simulation approaches can be found in [104, 105, 106] and [107], who compare platforms rather than approaches per se.

Prominent applications are various kinds of controlled systems. These seem predestined as hybrid systems due to their common representation by a continuous time system with a discrete control [108, 79, 101]. Specific applications range from power systems [102, 105], networked control systems [104, 107], voltage distribution control ([109], tanks with controller [101, 79], room temperature control [108] and manufacturing systems [110] to cyberphysical systems in general [91].

Especially developed frameworks are FIDE by [90], SAHISim by [68] (see also Section 4.1), CODIS by [98], an adaption of the Crescenco tool to combine Overture and 20-sim by [100], and a systematic approach for multi-level simulation by [110]. [102] consider the EPOCHS and GECO framework in their review of simulation methods of both communication and power systems. [111] propose a conceptual structuration of co-simulation frameworks consisting of the following five generic layers: conceptual (generic structure), semantic (interaction), syntactic (formalization), dynamic (execution, synchronization), and technical (implementation details, evaluation).

Recent developments in particular are utilizing the FMI in their solution approaches for hybrid systems [109, 68, 108, 75, 90, 79, 95, 103]. As the FMI by itself proves insufficient to satisfy requirements for hybrid co-simulation (see f.i. [112], [108] and cf. [75]), proposals for extensions to the FMI standard are given in [94, 91].

4.3 Coupled simulation of FEM models

Co-simulation of FEM models amongst themselves is covered by ([113, 114]). Others couple FEM with other models, such as multibody ([115]), BEM ([116]) or circuit models ([117]). The developments focus on specific kinds of applications such as fluid-structure interaction ([118, 116]), electro-thermal systems ([119, 117]) or vehicle dynamics ([115]). Most approaches in this area of application are either plainly sequential ([117, 118, 116]) or iterative ones ([119, 115, 113]).

4.4 Application-specific research

Many publications describe very specific applications that do not necessarily offer potential to aid general developments. Nevertheless, examples such as coupling methods for flow simulation [120], the development of a Functional Mockup Unit (FMU) in EnergyPlus for loose coupling co-simulation of Jacobi type with an EnergyPlus master [121], coupling MATLAB/Simulink with GENSYS using TCP/IP and S-functions for the simulation of rail traction vehicles [122], co-simulation in real-time hardware applications [123], or a survey on the state of the art in process-machine interactions focusing on metal-working processes [124] convey an idea of the vast field of areas profiting from the concept of co-simulation.

5 Partitioned Multirate Schemes

The introduction of partitioned multirate schemes is motivated by dividing stiff systems of ordinary differential equations into an active and latent part depending on the time constants of the respective subsystems: The active parts need to be integrated with a small step size, the latent parts with a comparatively large step size which is also used as macro time step. Stiffness is thus isolated in the latent parts which can therefore be integrated with an implicit algorithm while the active subsystems can be solved with an explicit solver [125, 126], incorporated in one partitioned solver algorithm. This way, computational effort can be reduced up to 90% [127].

According to [128], applying a multirate scheme is sensible if

- the systems (1) are weakly coupled, meaning $\left\|\frac{\partial f_1}{\partial \boldsymbol{x}_2}\right\| \ll \left\|\frac{\partial f_1}{\partial \boldsymbol{x}_1}\right\|$ and $\left\|\frac{\partial f_2}{\partial \boldsymbol{x}_1}\right\| \ll \left\|\frac{\partial f_2}{\partial \boldsymbol{x}_2}\right\|$
- "the activity levels are widely separated", meaning the micro steps are much smaller than the macro steps
- the activity is concentrated on a small part, meaning there are much less subcircuits in the active system

While the division into a system like (1) is mostly done “by hand” or even assumed to be given initially [127, 129], several approaches include automatic partitioning of the system (depending on step size comparisons, asymptotic behavior, precision of extrapolated values or error estimates), which is sometimes renewed after every macro step [125, 130, 131].

The regarded multirate schemes range from one-step (f.i. [132], who are co-simulating partitioned electrical networks with a w-multirate method or [133], who present an adaptive multirate strategy with a two-stage second-order Rosenbrock method) to multi-step methods [134] and variants [135] including slowest first [136, 125, 129], fastest first [131] and compound methods [126, 129]. Adaptive approaches have been developed by [133] and [137], who control the step size of both micro and macro steps.

Detailed investigations on stability properties of different multirate schemes are found in [138, 134, 129, 139]. [133] conduct error estimates for Rosenbrock methods depending on integration and interpolation orders which are utilized in the adaption of the step size. [140] conduct error analysis for the BDF compound-fast multirate method presented in [129].

While hierarchical structures have up to now been mostly neglected in classical integrate-and-collaborate co-simulation (cf. Section 6.1 from [2]), they have long been introduced for partitioned schemes: [134] already consider three different step sizes. [136] do not restrict the number of levels as long as these show “triangular” dependencies – i.e., equations can be ordered so that System 1 does not depend on values from any other subsystems, System 2 may only depend on values from Systems 1 and 2 and so on. [128] develops an approach suitable for an arbitrary number of activity levels that does not actually restrict dependencies but acknowledges that the partitioning only makes sense for weakly coupled systems, meaning relatively small magnitudes of dependencies (measured by the derivative of the right hand side by the respective state variables, see below).

An apt summary of limitations of multirate methods has been formulated by [29]:

> (...) if the mechatronic system is modelled according to the weakly coupled strategy, these multirate integration methods cannot be applied directly due to their particular features:

1. They introduce modifications in the integration schemes, something that is not possible in commercial off-the-shelf modeling and simulation tools used for weakly coupled co-simulation. For example, the aforementioned block diagram simulators and multibody system simulation packages offer their own set of integration schemes that cannot be modified.

2. They assume that the coarse and refined time-grids are equidistant and synchronized, which means that the large stepsize H is a multiple of the small stepsize h. This condition cannot be guaranteed in weakly coupled co-simulations if one or more subsystems are integrated with a variable time-step integrator, since the stepsize control algorithms of the different commercial simulation environments cannot be synchronized.

3. They mitigate the unstable behavior caused by the explicit extrapolation of some equation terms by introducing implicit schemes, which involve some kind of iterative process. Again, off-the-shelf simulation tools such as block diagram simulators do not allow this kind of iteration with other simulation tools.

6 General Information

In this section, general strategies for coupling methods ([34]), validation and verification of co-simulation ([55]) and results from a survey by [141] on the state of the art in co-simulation, including challenges in discrete event, continuous time and hybrid co-simulation, are summarized.

[34] present guidelines for an effective gluing algorithm, aiming to "execute coupled system simulation without sacrificing the integrity of subsystem modeling and solution and to maintain the efficacy of the overall results." They state that such an algorithm has to be

- *Sticky*: The inter-connection relations between subdomains should be well satisfied, i.e. coupling between subdomains should be resolved and captured.
- *Green*: It should not contaminate subdomain solution strategy. The integrity of the individual model and solution methods should be maintained. Minimum modification of the original solution scheme is desired.
- *Inexpensive*: The overhead should be minimized.
- *Pretty*: The results should be pretty; that is, the overall solution should be numerically correct within the bounds of the desired accuracy. [34]

Chapter 6 of [55] is dedicated to validation and verification of co-simulation. In general, validation is about whether the conceptual model describes the regarded system accurately, verification about the correct implementation and simulation of the conceptual model. [55] has verified her co-simulation by: static verification (structural properties of the code) and dynamical verification (exact synchronization and data transfer tested by varying of time constants).

Validation for coupled simulation is tricky as for different simulation tools ofttimes only different validation approaches exist and comparison with mono-simulation might not be expedient as modeling and simulation of the same system in only one (and hence different for at least one subsystem) simulator could yield different results due to the differences in the simulation tools. [55] apply a method based on inter-model comparison, using only one simulator for mono- and co-simulation.

[141] provide a survey on state-of-the-art techniques for co-simulation, starting by the introduction of a formalization similar to DEVS ([76]). As challenges specific to DE co-simulation, [141] name causality (especially for parallel execution with the possibility of rollback), determinism and confluence (the same results for all possible interleavings of executions), dynamic structure (varying dependencies), and distribution. Fulfillment of algebraic constraints and algebraic loops (closed-loop feed-through in input-output dependencies), which are of special interest for coupled DAE systems, are named as typical challenges in CT co-simulation next to consistent initialization, compositional convergence (error control), compositional stability, compositional continuity (discontinuities in input trajectories due to extrapolation), and real-time constraints. Formalization of hybrid (CT/DE) co-simulation is considered a non-trivial task and thus not given specifically. However, the idea is explained and specific challenges are given, the latter being semantic adaptation (the choice of wrappers depends on the

co-simulation scenario); predictive step sizes (fixed step sizes will miss events, adaptive approaches require detailed information on the subsystems); event location (related to step size prediction, requires information for prediction or rollback functionality); discontinuity identification; discontinuity handling (re-initializing might cause others and not terminate, energy conservation has to be respected); algebraic loops, legitimacy (infinite events at the same time step), and zeno behavior (infinite, consecutive events in ever decreasing intervals but in a bounded time frame, hard to detect in hybrid co-simulation); stability (issues of different origin; further analysis required); theory of DE approximated states (error bounds for the DE part) and establishing a standard for hybrid co-simulation. A taxonomy of a broad selection of literature on co-simulation has revealed the following most observed non-functional requirements: accuracy, protection of intellectual property and performance. Extensibility is among the least observed. Within framework requirements, least observed are dynamic structure co-simulation, interactive visualization, multi-rate, algebraic coupling, and partial/full strong coupling support. In general, they find that there is a lack of research in methods which are both DE and CT based and in leveraging features from simulation units.

7 Conclusion

This paper has given insights on various developments in the area of multirate and co-simulation, therein common methods, standards and frameworks. While there are broad areas of application and research, most investigations and developments are specialized on a certain kind of underlying equation system and may demand restrictions on the manner of coupling. This is not altogether surprising, as special problems come with specialized demands on their solution, which leads us to the most important conclusion to be drawn from this survey: that the choice for the one or the other method cannot be made globally but depends on the underlying system, the status of model development, know-how and interdisciplinarity of the team of developers.

This holds true for selecting special coupling algorithms – see f.i. [43], who show that depending on the system, even higher order extrapolation or higher macro step sizes can yield more stable results – as well as determining whether or not to approach a problem via co-simulation at all: For instance, the disadvantage mentioned in [80] that integration of hybrid aspects on the semantic level is not possible with their chosen co-simulation approach (in comparison to a DEV&DESS-based solution) could for some use cases be seen as advantage, as co-simulation does not require detailed insight and understanding of the partial models' description but *allows* them to be developed independently by experts in the corresponding domain or field. With regard to the additional capabilities or intrusions into subsystem simulators which would be required for rollbacks in co-simulation, this is a minor requirement of insight in comparison to the renewed formalization of every participating model.

In addition, we can observe that, while sensible for the reasons given above, restriction of investigations to systems fulfilling certain requirements holds a few risks: There exist several software tools allowing the more or less easy coupling of certain simulators. Unfortunately, these are often used without further investigation on the consequences regarding numerical stability – such as, for example, testing the system and used algorithms for the requirements necessary to guarantee stability. This, among others, holds true for hierarchical or nested co-simulation, which is allowed by some tools and even, although scarcely, performed, but has only recently been investigated regarding consistency and stability [142, 2].

The restriction to cases with special requirements also leaves a lot of unexploited methods for further investigations. Likewise does the pressing topic of hybrid co-simulation, for which promising developments are in progress in the research groups around the authors of [75, 91, 143]. We conclude with the observation in the words of [29] that "it is not possible to find an optimal general purpose co-simulation method", which leaves co-simulation as ever present topic of interest with plenty of open research questions to be addressed in the future.

References

[1] Hafner I, Popper N. On the Terminology and Structuring of Co-simulation Methods. In: *Proceedings of the 8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, EOOLT '17. Weßling, Germany: ACM. 2017; pp. 67–76.

[2] Hafner I. Cooperative and Multirate Simulation: Analysis, Classification and New Hierarchical

Approaches. phd thesis, TU Wien, Vienna, Austria. 2021. (submitted for publication).

[3] Hofer E. A Partially Implicit Method for Large Stiff Systems of ODEs with Only Few Equations Introducing Small Time-Constants. *SIAM Journal on Numerical Analysis*. 1976;13(5):645–663.

[4] Andrus J. Numerical Solution of Systems of Ordinary Differential Equations Separated into Subsystems. *SIAM Journal on Numerical Analysis*. 1979; 16(4):605–611.

[5] Gear CW, Wells DR. Multirate linear multistep methods. *BIT Numerical Mathematics*. 1984; 24(4):484–502.

[6] Jackson KR. A survey of Parallel Numerical Methods for Initial Value Problems for Ordinary Differential Equations. *IEEE Transactions on Magnetics*. 1991; 27(5):3792–3797.

[7] White J, Odeh F, Sangiovanni-Vincentelli A, Ruehli A. Waveform Relaxation: Theory and Practice. *Technical Report M85/65*, EECS Department, University of California, Berkeley. 1985.

[8] Knorr S. Multirate-Verfahren in der Co-Simulation gekoppelter dynamischer Systeme mit Anwendung in der Fahrzeugdynamik. Diploma Thesis, Universität Ulm, Ulm, Germany. 2002. Last accessed 1 Oct 2021.

[9] Gu B, Gordon BW, Asada HH. Co-simulation of coupled dynamic subsystems: a differential-algebraic approach using singularly perturbed sliding manifolds. In: *American Control Conference, 2000. Proceedings of the 2000*, vol. 2. 2000; pp. 757–761 vol.2.

[10] Kübler R, Schiehlen W. Two Methods of Simulator Coupling. *Mathematical and Computer Modelling of Dynamical Systems*. 2000;6(2):93–113.

[11] Gu B, Asada HH. Co-Simulation of Algebraically Coupled Dynamic Subsystems Without Disclosure of Proprietary Subsystem Models. *Journal of Dynamic Systems, Measurement, and Control*. 2004; 126(1):1–13.

[12] Gu B. Co-simulation of algebraically coupled dynamic subsystems. Thesis, Massachusetts Institute of Technology. 2001.

[13] Glumac S, Kovacic Z. Calling Sequence Calculation for Sequential Co-simulation Master. In: *Proceedings of the 2018 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '18. New York, NY, USA: ACM. 2018; pp. 157–160. Event-place: Rome, Italy.

[14] Stecken J, Lenkenhoff K, Kuhlenkötter B. Classification method for an automated linking of models in the co-simulation of production systems. *Procedia CIRP*. 2019;81:104–109.

[15] Lelarasmee E, Ruehli A, Sangiovanni-Vincentelli A. The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1982;1(3):131–145.

[16] Arnold M, Günther M. Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems. *BIT Numerical Mathematics*. 2001;41(1):1–25.

[17] Rathinam M, Petzold L. Dynamic Iteration Using Reduced Order Models: A Method for Simulation of Large Scale Modular Systems. *SIAM Journal on Numerical Analysis*. 2002;40(4):1446–1474.

[18] Ebert F. Convergence of relaxation methods for coupled systems of ODEs and DAEs. 2004;Accessed on 11-08-2021.
URL
`https://opus4.kobv.de/opus4-matheon/frontdoor/index/index/docId/177`

[19] Tomulik P, Fraczek J. Simulation of multibody systems with the use of coupling techniques: a case study. *Multibody System Dynamics*. 2011; 25(2):145–165.

[20] Sicklinger S, Belsky V, Engelmann B, Elmqvist H, Olsson H, Wüchner R, Bletzinger KU. Interface Jacobian-based Co-Simulation. *International Journal for Numerical Methods in Engineering*. 2014; 98(6):418–444.

[21] Sicklinger S, Lerch C, Wüchner R, Bletzinger KU. Fully coupled co-simulation of a wind turbine emergency brake maneuver. *Journal of Wind Engineering and Industrial Aerodynamics*. 2015; 144:134–145.

[22] Schöps S. Multiscale Modeling and Multirate Time-Integration of Field/Circuit Coupled Problems. Ph.D. thesis, Universität Wuppertal, Wuppertal. 2011.

[23] Ebert F. On Partitioned Simulation of Electrical Circuits using Dynamic Iteration Methods. Doctoral thesis, Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Berlin. 2008.

[24] Busch M. *Zur effizienten Kopplung von Simulationsprogrammen*. Kassel University Press. 2012.

[25] Schmoll R. Co-Simulation und Solverkopplung. Buch, Kassel Univ. Press. 2015.

[26] Völker L. *Untersuchung des Kommunikationsintervalls bei der gekoppelten Simulation*. No. Bd. 6 in Karlsruher Schriftenreihe

Fahrzeugsystemtechnik. Karlsruhe: KIT Scientific Publ. 2011.

[27] Benedikt M, Stippel H, Watzenig D. An Adaptive Coupling Methodology for Fast Time-Domain Distributed Heterogeneous Co-Simulation. In: *SAE 2010 World Congress & Exhibition*. 2010; .

[28] Liang S, Zhang H, Wang H. Combinative Algorithms for the Multidisciplinary Collaborative Simulation of Complex Mechatronic Products Based on Major Step and Convergent Integration Step. *Chinese Journal of Mechanical Engineering*. 2011;24(03):355.

[29] González F, Naya Mn, Luaces A, González M. On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics. *Multibody System Dynamics*. 2011; 25(4):461–483.

[30] Schierz T, Arnold M. MODELISAR: Innovative numerische Methoden bei der Kopplung von multidisziplinären Simulationsprogrammen. In: *Tagungsband ASIM-Konferenz STS/GMMS 2011*. ZHAW Winterthur, Schweiz: Shaker. 2011; .

[31] Jia Z, Leimkuhler B. A parallel multiple time-scale reversible integrator for dynamics simulation. *Future Generation Computer Systems*. 2003;19(3):415–424.

[32] Featherstone R. A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 1: Basic Algorithm. *The International Journal of Robotics Research*. 1999;18(9):867–875.

[33] Featherstone R. A Divide-and-Conquer Articulated-Body Algorithm for Parallel O(log(n)) Calculation of Rigid-Body Dynamics. Part 2: Trees, Loops, and Accuracy. *The International Journal of Robotics Research*. 1999;18(9):876–892.

[34] Tseng FC, Hulbert G. A Gluing Algorithm for Network-Distributed Multibody Dynamics Simulation. *Multibody System Dynamics*. 2001;6(4):377–396.

[35] Schweizer B, Li P, Lu D. Explicit and Implicit Cosimulation Methods: Stability and Convergence Analysis for Different Solver Coupling Approaches. *Journal of Computational and Nonlinear Dynamics*. 2015;10(5):051007.

[36] Tseng FC, Ma ZD, Hulbert GM. Efficient numerical solution of constrained multibody dynamics systems. *Computer Methods in Applied Mechanics and Engineering*. 2003;192(3–4):439–472.

[37] Wang J, Ma ZD, Hulbert GM. A Gluing Algorithm for Distributed Simulation of Multibody Systems. *Nonlinear Dynamics*. 2003;34(1):159–188.

[38] Wang J, Ma ZD, Hulbert GM. A Distributed Mechanical System Simulation Platform Based on a “Gluing Algorithm”. *Journal of Computing and Information Science in Engineering*. 2005;5(1):71.

[39] Rustin C, Verlinden O, Bombled Q. A Cosimulation T-T Procedure Gluing Subsystems in Multibody Dynamics Simulations. ASME. 2009; pp. 83–92.

[40] Schweizer B, Lu D. Stabilized index-2 co-simulation approach for solver coupling with algebraic constraints. *Multibody System Dynamics*. 2014; 34(2):129–161.

[41] Schweizer B, Lu D. Semi-implicit co-simulation approach for solver coupling. *Archive of Applied Mechanics*. 2014;84(12):1739–1769.

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

[43] Schweizer B, Li P, Lu D. Implicit co-simulation methods: Stability and convergence analysis for solver coupling approaches with algebraic constraints. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*. 2016;96(8):986–1012.

[44] Schweizer B, Li P, Lu D, Meyer T. Stabilized implicit co-simulation methods: solver coupling based on constitutive laws. *Archive of Applied Mechanics*. 2015; 85(11):1559–1594.

[45] Matthies HG, Steindorf J. Partitioned but strongly coupled iteration schemes for nonlinear fluid–structure interaction. *Computers & Structures*. 2002; 80(27–30):1991–1999.

[46] Matthies HG, Steindorf J. Strong Coupling Methods. In: *Analysis and Simulation of Multifield Problems*, edited by Wendland PW, Efendiev PM, no. 12 in Lecture Notes in Applied and Computational Mechanics, pp. 13–36. Springer Berlin Heidelberg. 2003;.

[47] Matthies HG, Niekamp R, Steindorf J. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*. 2006; 195(17–18):2028–2049.

[48] Trčka M, Wetter M, Hensen J. Comparison of co-simulation approaches for building and HVAC/R system simulation. In: *Proc. of the 10th IBPSA Conference*. Beijing, China. 2007; pp. 1418–1425.

[49] Trčka M, Hensen JL, Wetter M. Co-simulation of innovative integrated HVAC systems in buildings.

Journal of Building Performance Simulation. 2009; 2(3):209–230.

[50] Pühringer C. Analysis of Coupling Strategies and Protocols for Co-Simulation. Diplomarbeit, TU Wien, Wien. 2017.

[51] González F, González M, Mikkola A. Efficient coupling of multibody software with numerical computing environments and block diagram simulators. *Multibody System Dynamics*. 2010; 24(3):237–253.

[52] Arnold M. Multi-Rate Time Integration for Large Scale Multibody System Models. In: *IUTAM Symposium on Multiscale Problems in Multibody System Contacts*, edited by Eberhard P, no. 1 in IUTAM Bookseries, pp. 1–10. Springer Netherlands. 2007;.

[53] Arnold M, Hante S, Köbis MA. Error analysis for co-simulation with force-displacement coupling. *PAMM*. 2014;14(1):43–44.

[54] Schmoll R, Schweizer B. Convergence Study of Explicit Co-Simulation Approaches with Respect to Subsystem Solver Settings. *PAMM*. 2012;12(1):81–82.

[55] Trčka M. Cosimulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven. 2008.

[56] Zhang H, Liang S, Song S, Wang H. Truncation error calculation based on Richardson extrapolation for variable-step collaborative simulation. *Science China Information Sciences*. 2011;54(6):1238–1250.

[57] Arnold M, Clauss C, Schierz T. Error Analysis and Error Estimates for Co-Simulation in FMI for Model Exchange and Co-Simulation V2.0. *Archive of Mechanical Engineering*. 2013;LX(1).

[58] Glumac S, Kovacic Z. Relative Consistency and Robust Stability Measures for Sequential Co-simulation. 2019; pp. 197–206.

[59] Bartel A, Brunk M, Schöps S. On the convergence rate of dynamic iteration for coupled problems with multiple subsystems. *Journal of Computational and Applied Mathematics*. 2014;262:14–24.

[60] Arnold M, Burgermeister B, Führer C, Hippmann G, Rill G. Numerical methods in vehicle system dynamics: state of the art and current developments. *Vehicle System Dynamics*. 2011;49(7):1159–1207.

[61] Arnold M. Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models. *Journal of Computational and Nonlinear Dynamics*. 2010;5(3):031003.

[62] Schierz T, Arnold M. Stabilized overlapping modular time integration of coupled differential-algebraic equations. *Applied Numerical Mathematics*. 2012; 62(10):1491–1502.

[63] Larsson J, Krus P. Stability Analysis of Coupled Simulation. vol. Dynamic Systems and Control, Volumes 1 and 2 of *ASME International Mechanical Engineering Congress and Exposition*. 2003; pp. 861–868.

[64] Sadjina S, Pedersen E. Energy Conservation and Coupling Error Reduction in Non-Iterative Co-Simulations. *arXiv:160605168 [cs]*. 2016;ArXiv: 1606.05168.

[65] Viel A. Implementing stabilized co-simulation of strongly coupled systems using the Functional Mock-up Interface 2.0. 2014; pp. 213–223.

[66] Bartel A, Brunk M, Günther M, Schöps S. Dynamic Iteration for Coupled Problems of Electric Circuits and Distributed Devices. *SIAM Journal on Scientific Computing*. 2013;35(2):B315–B335.

[67] Dahmann JS, Fujimoto RM, Weatherly RM. The Department of Defense High Level Architecture. In: *Proceedings of the 29th conference on Winter simulation - WSC '97*. Atlanta, Georgia, United States: ACM Press. 1997; pp. 142–149.

[68] Awais MU. Distributed hybrid co-simulation. Ph.D. thesis, TU Wien, Vienna, Austria. 2015.

[69] Blockwitz T, Otter M, Akesson J, Arnold M, Clauss C, Elmqvist H, Friedrich M, Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A. Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. 2012; pp. 173–184.

[70] Modelica Association. Functional Mock-up Interface for Model Exchange and Co-Simulation. `https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI_for_ModelExchange_and_CoSimulation_v2.0.pdf`. 2014. Accessed on 2017-10-19.

[71] Schweiger G, Gomes C, Engel G, Hafner I, Schoegg JP, Posch A, Nouidui T. Functional Mock-up Interface: An empirical survey identifies research challenges and current barriers. 2019; pp. 138–146.

[72] Broman D, Brooks C, Greenberg L, Lee EA, Masin M, Tripakis S, Wetter M. Determinate Composition of FMUs for Co-simulation. In: *Proceedings of the Eleventh ACM International Conference on Embedded Software*, EMSOFT '13. Piscataway, NJ, USA: IEEE Press. 2013; pp. 2:1–2:12. Event-place: Montreal, Quebec, Canada.

[73] Tripakis S. Bridging the semantic gap between heterogeneous modeling formalisms and FMI. In: *2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. Samos, Greece: IEEE. 2015; pp. 60–69.

[74] Gomes C, Thule C, Lúcio L, Vangheluwe H, Larsen PG. Generation of Co-simulation Algorithms Subject to Simulator Contracts. Oslo, Norway. 2019; p. 15.

[75] Broman D, Greenberg L, Lee EA, Masin M, Tripakis S, Wetter M. Requirements for Hybrid Cosimulation Standards. In: *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15. New York, NY, USA: ACM. 2015; pp. 179–188.

[76] Zeigler BP, Praehofer H, Kim TG. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. San Diego: Academic Press, 2nd ed. 2000.

[77] Preyser FJ. An approach to develop a user friendly way of implementing DEV&DESS models in powerDEVS. Thesis. 2015.

[78] Deatcu C, Pawletta T. A Qualitative Comparison of Two Hybrid DEVS Approaches. *SNE Simulation Notes Europe*. 2012;22(1):15–24.

[79] Camus B, Galtier V, Caujolle M. Hybrid Co-simulation of FMUs using DEV DESS in MECSYCO. In: *2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS)*. 2016; pp. 1–8.

[80] Heinzl B, Raich P, Preyser F, Kastner W. Simulation-based Assessment of Energy Efficiency in Industry: Comparison of Hybrid Simulation Approaches. *IFAC-PapersOnLine*. 2018; 51(2):689–694.

[81] Wetter M. Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation*. 2011;4(3):185–203.

[82] Zhang Z, Eyisi E, Koutsoukos X, Porter J, Karsai G, Sztipanovits J. Co-simulation framework for design of time-triggered cyber physical systems. *Simulation Modelling Practice and Theory*. 2014;43:16–33.

[83] Ferreira PAF, Esteves EF, Rossetti RJF, Oliveira EC. A Cooperative Simulation Framework for Traffic and Transportation Engineering. In: *Cooperative Design, Visualization, and Engineering*, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2008; pp. 89–97.

[84] Friedrich M. Parallel Co-Simulation for Mechatronic Systems. Ph.D. thesis, Technische Universität München. 2011.

[85] Karsai G, Sztipanovits J. Model-Integrated Development of Cyber-Physical Systems. In: *Software Technologies for Embedded and Ubiquitous Systems*, edited by Brinkschulte U, Givargis T, Russo S, Lecture Notes in Computer Science. Springer Berlin Heidelberg. 2008; pp. 46–54.

[86] Galtier V, Vialle S, Dad C, Tavella JP, Lam-Yee-Mui JP, Plessis G. FMI-based distributed multi-simulation with DACCOSIM. In: *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, DEVS '15. San Diego, CA, USA: Society for Computer Simulation International. 2015; pp. 39–46.

[87] Ben Khaled A, Duval L, Ben Gaid M, Simon D. Context-based polynomial extrapolation and slackened synchronization for fast multi-core simulation using FMI. Lund, Sweden: inköping University Electronic Press,. 2014; pp. 225–234.

[88] Wang K, Siebers PO, Robinson D. Towards Generalized Co-simulation of Urban Energy Systems. *Procedia Engineering*. 2017;198:366–374.

[89] Thule C, Palmieri M, Gomes C, Lausdahl K, Macedo HD, Battle N, Larsen PG. Towards Reuse of Synchronization Algorithms in Co-simulation Frameworks. In: *Software Engineering and Formal Methods*, edited by Camara J, Steffen M. Cham: Springer International Publishing. 2020; pp. 50–66.

[90] Cremona F, Lohstroh M, Tripakis S, Brooks C, Lee EA. FIDE: An FMI Integrated Development Environment. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16. New York, NY, USA: ACM. 2016; pp. 1759–1766. Event-place: Pisa, Italy.

[91] Cremona F, Lohstroh M, Broman D, Lee EA, Masin M, Tripakis S. Hybrid co-simulation: it's about time. *Software & Systems Modeling*. 2019;18(3):1655–1679.

[92] Mosterman PJ. An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. In: *Hybrid Systems: Computation and Control*, edited by Goos G, Hartmanis J, van Leeuwen J, Vaandrager FW, van Schuppen JH, vol. 1569, pp. 165–177. Berlin, Heidelberg: Springer Berlin Heidelberg. 1999;.

[93] Zhang F, Yeddanapudi M, Mosterman PJ. Zero-Crossing Location and Detection Algorithms For Hybrid System Simulation. *IFAC Proceedings Volumes*. 2008;41(2):7967–7972.

[94] Cremona F, Lohstroh M, Broman D, Natale MD, Lee EA, Tripakis S. Step revision in hybrid Co-simulation with FMI. In: *2016 ACM/IEEE International*

Conference on Formal Methods and Models for System Design (MEMOCODE). 2016; pp. 173–183.

[95] Thule C, Gomes C, Deantoni J, Larsen PG, Brauer J, Vangheluwe H. Towards the Verification of Hybrid Co-simulation Algorithms. In: *Software Technologies: Applications and Foundations*, edited by Mazzara M, Ober I, Salaün G, vol. 11176, pp. 5–20. Cham: Springer International Publishing. 2018;.

[96] Barros FJ. Chattering Avoidance in Hybrid Simulation Models: A Modular Approach Based on the HyFlow Formalism. In: *Proceedings of the Symposium on Theory of Modeling & Simulation*, TMS/DEVS '17. San Diego, CA, USA: Society for Computer Simulation International. 2017; pp. 15:1–15:12.

[97] Van Mierlo S, Gomes C, Vangheluwe H. Explicit Modelling and Synthesis of Debuggers for Hybrid Simulation Languages. In: *Proceedings of the Symposium on Theory of Modeling & Simulation*, TMS/DEVS '17. San Diego, CA, USA: Society for Computer Simulation International. 2017; pp. 4:1–4:12.

[98] Gheorghe L. Continuous/Discrete Co-Simulation Interfaces from Formalization to Implementation. phd, École Polytechnique de Montréal. 2009.

[99] Barros FJ. Semantics of Dynamic Structure Event-based Systems. In: *Proceedings of the Second International Conference on Distributed Event-based Systems*, DEBS '08. New York, NY, USA: ACM. 2008; pp. 245–252.

[100] Fitzgerald J, Larsen PG, Verhoef M, eds. *Collaborative Design for Embedded Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2014. DOI: 10.1007/978-3-642-54118-6.

[101] Tudoret S, Nadjm-Tehrani S, Benveniste A, Strömberg JE. Co-Simulation of Hybrid Systems: Signal-Simulink. In: *Formal Techniques in Real-Time and Fault-Tolerant Systems*, edited by Joseph M. Berlin, Heidelberg: Springer Berlin Heidelberg. 2000; pp. 134–151.

[102] Tong H, Ni M, Yu W, Li Y. Reviews and perspectives of hybrid system simulation for power and communication. In: *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*. 2014; pp. 302–306.

[103] Farkas R, Bergmann G, Horváth k. Adaptive Step Size Control for Hybrid CT Simulation without Rollback. In: *Proceedings of the 13th International Modelica Conference*, vol. 157 of *Linköping Electronic Conference Proceedings*. Regensburg, Germany: Linköping University Electronic Press. 2019; p. 10.

[104] Quaglia D, Muradore R, Bragantini R, Fiorini P. A SystemC/Matlab co-simulation tool for networked control systems. *Simulation Modelling Practice and Theory*. 2012;23:71–86.

[105] Palensky P, Widl E, Elsheikh A. Simulating Cyber-Physical Energy Systems: Challenges, Tools and Methods. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*. 2014; 44:318 – 326.

[106] Heinzl B. Hybrid Modeling of Production Systems: Co-simulation and DEVS-based Approach. Diplomarbeit, TU Wien, Wien. 2016.

[107] Li W, Zhang X, Li H. Co-simulation platforms for co-design of networked control systems: An overview. *Control Engineering Practice*. 2014;23:44–56.

[108] Widl E, Judex F, Eder K, Palensky P. FMI-based co-simulation of hybrid closed-loop control system models. In: *2015 International Conference on Complex Systems Engineering (ICCSE)*. 2015; pp. 1–6.

[109] Savicks V, Butler M, Colley J. Co-simulating event-B and Continuous Models via FMI. In: *Proceedings of the 2014 Summer Simulation Multiconference*, SummerSim '14. Society for Computer Simulation International. 2014; pp. 37:1–37:8. Event-place: Monterey, California.

[110] Thiede S, Schönemann M, Kurle D, Herrmann C. Multi-level simulation in manufacturing companies: The water-energy nexus case. *Journal of Cleaner Production*. 2016;139:1118–1127.

[111] Nguyen V, Besanger Y, Tran Q, Nguyen T. On Conceptual Structuration and Coupling Methods of Co-Simulation Frameworks in Cyber-Physical Energy System Validation. *Energies*. 2017;10(12):1977.

[112] Schweiger G, Gomes C, Engel G, Hafner I, Schoeggl J, Posch A, Nouidui T. An empirical survey on co-simulation: Promising standards, challenges and research needs. *Simulation Modelling Practice and Theory*. 2019;95:148–163.

[113] Ibrahimbegović A, Markovič D. Strong coupling methods in multi-phase and multi-scale modeling of inelastic behavior of heterogeneous structures. *Computer Methods in Applied Mechanics and Engineering*. 2003;192(28–30):3089–3107l.

[114] Esgandari M, Olatunbosun O. Implicit–explicit co-simulation of brake noise. *Finite Elements in Analysis and Design*. 2015;99:16–23.

[115] Mousseau CW, Laursen TA, Lidberg M, Taylor RL. Vehicle dynamics simulations with coupled multibody and finite element models. *Finite Elements in Analysis and Design*. 1999;31(4):295–315.

[116] Felippa CA, Park K, Farhat C. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*. 2001; 190(24-25):3247–3270.

[117] Wünsche S, Clauss C, Schwarz P, Winkler F. Electro-thermal circuit simulation using simulator coupling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 1997;5(3):277–282.

[118] Farhat C, Lesoinne M. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*. 2000;182(3-4):499–515.

[119] van Petegem W, Geeraerts B, Sansen W, Graindourze B. Electrothermal simulation and design of integrated circuits. *IEEE Journal of Solid-State Circuits*. 1994; 29(2):143–146.

[120] Steinebach G, Rademacher S, Rentrop P, Schulz M. Mechanisms of coupling in river flow simulation systems. *Journal of Computational and Applied Mathematics*. 2004;168(1-2):459–470.

[121] Nouidui T, Wetter M, Zuo W. Functional mock-up unit for co-simulation import in EnergyPlus. *Journal of Building Performance Simulation*. 2014;7(3):192–202.

[122] Spiryagin M, Simson S, Cole C, Persson I. Co-simulation of a mechatronic system using Gensys and Simulink. *Vehicle System Dynamics*. 2012; 50(3):495–507.

[123] Stettinger G, Benedikt M, Horn M, Zehetner J. Modellbasierte Echtzeit-Co-Simulation: Überblick und praktische Anwendungsbeispiele. *e & i Elektrotechnik und Informationstechnik*. 2015; 132(4-5):207–213.

[124] Brecher C, Esser M, Witt S. Interaction of manufacturing process and machine tool. *CIRP Annals*. 2009;58(2):588–607.

[125] Günther M, Rentrop P. Partitioning and Multirate Strategies in Latent Electric Circuits. In: *Mathematical Modelling and Simulation of Electrical Circuits and Semiconductor Devices*, edited by Bank RE, Gajewski H, Bulirsch R, Merten K, ISNM International Series of Numerical Mathematics. Basel: Birkhäuser. 1994; pp. 33–60.

[126] Günther M, Kværnø A, Rentrop P. Multirate Partitioned Runge-Kutta Methods. *BIT Numerical Mathematics*. 2001;41(3):504–514.

[127] Rice JR. Split Runge-Kutta method for simultaneous equations. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*. 1960;64B(3):151.

[128] Striebel M. Hierarchical Mixed Multirating for Distributed Integration of DAE Network Equations in Chip Design. Ph.D. thesis, Bergische Universität Wuppertal. 2006.

[129] Verhoeven A, Guennouni AE, ter Maten EJW, Mattheij RMM. *A General Compound Multirate Method for Circuit Simulation Problems*, pp. 143–149. Berlin, Heidelberg: Springer Berlin Heidelberg. 2006;.

[130] Engstler C, Lubich C. Multirate extrapolation methods for differential equations with different time scales. *Computing*. 1997;58(2):173–185.

[131] Kvaernø¸ A, Rentrop P. Low Order Multirate Runge-Kutta Methods in Electric Circuit Simulation. 1999.

[132] Bartel A, Günther M. A multirate W-method for electrical networks in state-space formulation. *Journal of Computational and Applied Mathematics*. 2002; 147(2):411 – 425.

[133] Savcenco V, Hundsdorfer W, Verwer JG. A multirate time stepping strategy for stiff ordinary differential equations. *BIT Numerical Mathematics*. 2007; 47(1):137–155.

[134] Skelboe S, Andersen PU. Stability Properties of Backward Euler Multirate Formulas. *SIAM Journal on Scientific and Statistical Computing*. 1989; 10(5):1000–1009.

[135] Biesiadecki JJ, Skeel RD. Dangers of Multiple Time Step Methods. *Journal of Computational Physics*. 1993;109(2):318–328.

[136] Esposito J, Kumar V. Efficient dynamic simulation of robotic systems with hierarchy. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3. Seoul, South Korea: IEEE. 2001; pp. 2818–2823.

[137] Verhoeven A, Tasić B, Beelen TGJ, ter Maten EJW, Mattheij RMM. BDF Compound-Fast Multirate Transient Analysis with Adaptive Stepsize Control. *Journal of Numerical Analysis, Industrial and Applied Mathematics*. 2008;3(3-4):275–297.

[138] Gomm W. Stability analysis of explicit multirate methods. *Mathematics and Computers in Simulation*. 1981;23(1):34–50.

[139] Verhoeven A, Maten EJWT, Mattheij RMM, Tasić B. Stability analysis of the BDF Slowest-first multirate methods. *International Journal of Computer Mathematics*. 2007;84(6):895–923.

[140] Verhoeven A, Beelen TGJ, Guennouni AE, ter Maten EJW, Mattheij RMM, Tasić B. Error analysis of BDF Compound-fast multirate method for

differential-algebraic equations. *CASA-Report*. 2006; 06(10).

[141] Gomes C, Thule C, Broman D, Larsen PG, Vangheluwe H. Co-simulation: State of the art. *arXiv:170200686 [cs]*. 2017;ArXiv: 1702.00686.

[142] Hafner I, Popper N. Investigation on Stability Properties of Hierarchical Co-Simulation. In: *Proceedings ASIM SST 2020, 25. Symposium Simulationstechnik*, vol. 59 of *ARGESIM Report*. Online-Tagung: ARGESIM Verlag. 2020; pp. 41–48.

[143] Gomes C, Karalis P, Navarro-López EM, Vangheluwe H. Approximated Stability Analysis of Bi-modal Hybrid Co-simulation Scenarios. In: *Software Engineering and Formal Methods*, edited by Cerone A, Roveri M, vol. 10729, pp. 345–360. Cham: Springer International Publishing. 2018;.