Parallel Fast: An Efficient Coupling Approach for Co-Simulation with Different Coupling Step Sizes

Franz Holzinger[1,2] Klaus Schuch[2] Martin Benedikt[1] Daniel Watzenig[1,3]

[1]VIRTUAL VEHICLE Research GmbH, Austria,
{franz.holzinger,martin.benedikt,daniel.watzenig}@v2c2.at
[2]AVL List GmbH, Austria, {franz.holzinger,klaus.schuch}@avl.com
[3]Institute of Automation and Control, Graz University of Technology, Austria, daniel.watzenig@tugraz.at

Abstract

The primary task of co-simulation is the synchronization and exchange of data between the subsystems, e.g., FMUs, at certain coupling points. If the FMUs have different step sizes, the synchronization of the FMUs is often at the expense of the simulation duration of the co-simulation. The presented parallel fast scheduling algorithm is an effective approach to couple FMUs with different coupling step sizes. Therefore, synchronization intervals are introduced in which FMUs that finish their coupling step are synchronized. This allows a high performance of the coupling in terms of simulation duration. The higher performance compared to other scheduling algorithms is particularly evident in real-time applications, i.e., HiL simulations. However, the synchronization intervals are defined via a synchronization step size, which can be set independently to the coupling step sizes of the FMUs. This additional step size has a significant impact on the simulation accuracy. An extrapolation measure is introduced, which approximates the impact of the synchronization step size on the extrapolation error and thus on the simulation accuracy. Based on this, an optimization approach is presented, which derives the optimal synchronization step size to minimize the extrapolation measure.

parallel scheduling, synchronization step size, optimal step size

1 Introduction

In order to reduce development costs and development time, the focus in the industry has increasingly been placed on simulation in the recent decades. This led to a multitude of simulation environments to solve the engineering tasks in the different domains and applications. However, the different simulation tools often cover a specific area. In order to consider interactions across domains, it is necessary to integrate the specific models and tools into a combined simulation. In contrary to remodelling of the several specific models, which is cost or at least time intensive, co-simulation enables the direct integration of the individual subsystems and models, whereby coupling variables are exchanged at certain time steps to synchronize the subsystems as introduced in Kübler and Schiehlen (2000). Standardisations in the interface have reduced the technical effort to integrate subsystems as FMUs from different simulation environments, for more details see Blochwitz et al. (2012).

However, besides the technical implementation of the FMU[1] integration and data exchange, there are challenges in coupling and synchronization of the FMUs. Especially with non-iterative co-simulation, where coupling steps cannot be repeated, scheduling approaches, step-sizes and extrapolation techniques have a major impact on the simulation accuracy and the simulation duration.

In order to solve the causality problem between interacting FMUs extrapolation of coupling signals is needed. The most common representative is the ZOH (zero-order-hold) extrapolation, where the last known value of the coupling signals is used to determine the upcoming coupling step. The coupling imperfection resulting from the extrapolation can be handled with particular extrapolation and compensation techniques. For instance, a signal based extrapolation technique to compensate the energy losses caused by the coupling is discussed in Benedikt and Drenth (2019). In Haid et al. (2018) a model-based predictor corrector approach is introduced and a model-based pre-step stabilization technique is shown in Genser and Benedikt (2018).

In addition to the extrapolation techniques, the coupling step sizes, i.e., the defined time steps of the data exchange between the FMUs, contribute significantly to the simulation accuracy and the simulation duration. Evaluation and definition of the coupling step size based on the instantaneous frequency of the coupling signals is discussed in Benedikt, Watzenig, et al. (2013). Coupling error based adaptive step size approaches are analysed in Busch and Schweizer (2011) and Sadjina et al. (2016).

Beside them the scheduling influences the simulation accuracy and the simulation duration of the co-simulation. The scheduling can be categorized into two major groups: parallel and sequential coupling. Sequential coupling means that the FMUs are executed one after the other, which allows subsequent FMUs to access the results of

[1]For reasons of better understanding, the term FMU will be used for models and subsystems from now on. This is not a general restriction, all considerations are also valid for other model integrations.

FMUs that have already been executed. This reduces the extrapolations of the entire co-simulation and increases the simulation accuracy, but at the expense of the simulation duration. The simulation results strongly depend on the execution order of the FMUs. Approaches to identifying suitable execution sequences are discussed in Glumac and Kovacic (2018), F. Holzinger and Benedikt (2019), and Oakes et al. (2020).

With parallel coupling, the FMUs are executed simultaneously, which typically leads to shorter simulation durations than sequential co-simulation. However, the simulation accuracy typically decreases due to an increasing need of extrapolation. Nevertheless, for a performant simulation in terms of simulation time, e.g., for real-time applications, parallel coupling approaches are typically preferred.

However, not only the timing behaviour of the FMUs themselves defines the overall simulation duration. Especially if different coupling step sizes are used for the FMUs, dedicated scheduling algorithms are needed for the synchronization between the FMUs. These scheduling algorithms and their underlying synchronization strategy have an additional effect on the simulation duration and the simulation accuracy. For instance, Matlab Simulink uses a superior step size (solver step size) to synchronize the FMUs. This has the restriction, that the step sizes of the individual FMUs have to be a multiple of the superior step size, for more details see Matlab Simulink (2021). Especially with large differences in the step sizes of the FMUs, this can lead to unnecessarily increased synchronization efforts in terms of data exchange.

The presented parallel fast coupling approach also uses a superior step size, the so-called synchronization step size. However, the synchronization step size can be defined independently of the coupling step sizes of the FMUs, which enables a suitable data exchange between the FMUs. The parallel fast scheduling with the synchronization step size shows a performant simulation despite to different coupling step sizes. The selection of the synchronization step size has an effect on the data exchange and thus on the simulation accuracy. In order to achieve an suitable configuration, an optimization approach to define the synchronization step size based on an extrapolation assessment of the coupling signals is discussed.

This work is structured as follows: The next chapter introduces the parallel fast scheduling and discusses its execution behaviour. In the third chapter the extrapolation measure is introduced, which approximates the extrapolation error based on the topology and configuration of the co-simulation. Based on the extrapolation measure an optimization approach to identify the optimal synchronization step size for the parallel fast scheduling is derived in chapter four. A co-simulation example with four FMUs S_A, S_B, S_C, S_D is used to illustrate the properties of the parallel scheduling and the optimization approach. The topology of the co-simulation example is shown in Figure 1. The FMUs of the example are based on an example in Benedikt and Drenth (2019) and F. Holzinger and Benedikt (2019). All FMUs refer to FMU for co-simulation according to the FMI version 2.0. The parallel fast approach is available as a scheduling procedure in the AVL Co-Simulation platform Model.CONNECT™ for more information see Model.CONNECT™ (2020).

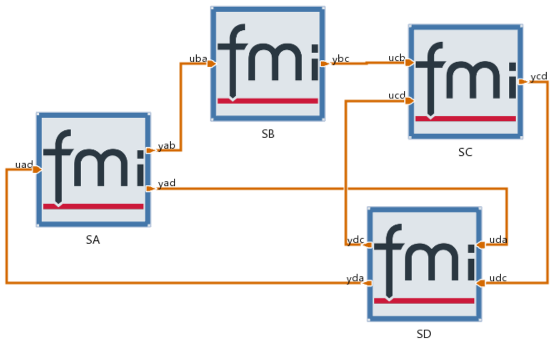


Figure 1. Co-Simulation topology of an example with four FMUs.

2 Parallel Fast Scheduling

Parallel scheduling algorithms execute the FMUs simultaneously. Due to the parallel execution, these approaches basically have a high performance in terms of simulation duration. Which makes them suitable for real-time applications, where the entire co-simulation must calculate faster than real-time. If the FMUs have the same coupling step sizes h_i, they have the identical simulation progress nh_i, i.e., simulation time $t_{s,i}$, after each coupling step n. With respect to the data exchange, the FMUs have to wait for each other after a coupling step. Neglecting the time for data exchange, the simulation duration and thus the real-time factor of the entire co-simulation is dominated by the slowest FMU and can be estimated as follows:

$$\hat{D} = \max\{d_i\}, \tag{1}$$

where d_i are the real-time factors of the individual FMUs S_i and $\hat{D}$ is the estimated real-time factor of the entire co-simulation. A real-time factor $\hat{D} = 1$ indicates that the co-simulation is running in real-time, i.e. the simulation time t_s is equal to the wall clock time t_w. If $\hat{D} < 1$, the co-simulation is faster than real time ($t_s < t_w$) and if $\hat{D} > 1$, the co-simulation is slower than real time ($t_s > t_w$). However, if different coupling step sizes of the FMUs are used, the simulation progress and thus the simulation time $t_{s,i}$ will differ for the individual FMUs S_i.

Depending on the coupling strategy the FMUs are synchronized at different time steps. Approaches, like the latest-first scheduling approach, where the several FMUs are synchronized after each coupling step, show a significant increasing of the simulation duration due to different coupling step sizes.

However, the presented parallel fast approach avoids such behaviour by introducing a global step size for synchronization, the so-called synchronization step size H.

DOI
10.3384/ecp21181649

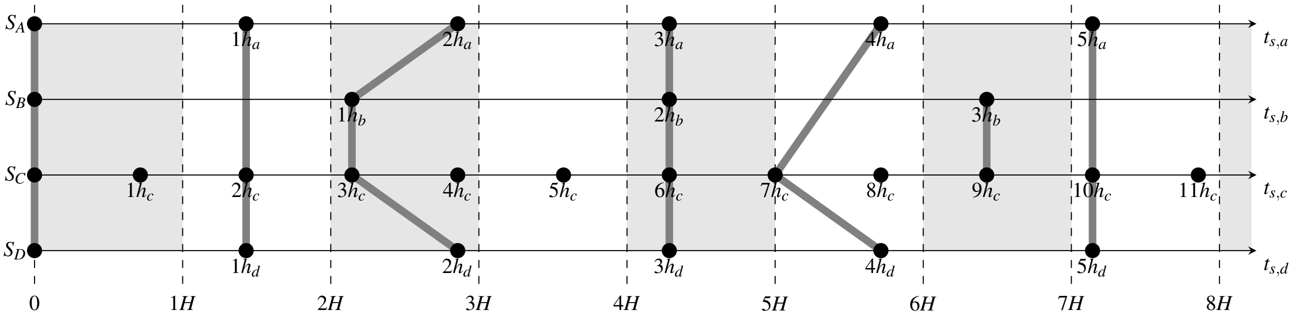


Figure 2. Execution sequence of the parallel fast scheduling.

The step size H defines synchronization intervals. If two or more FMUs end their coupling step within a synchronization interval $[\zeta H, (\zeta+1)H)$, data is exchanged between these systems. In general, this synchronizations condition for two FMUs S_i and S_j can be written as follows:

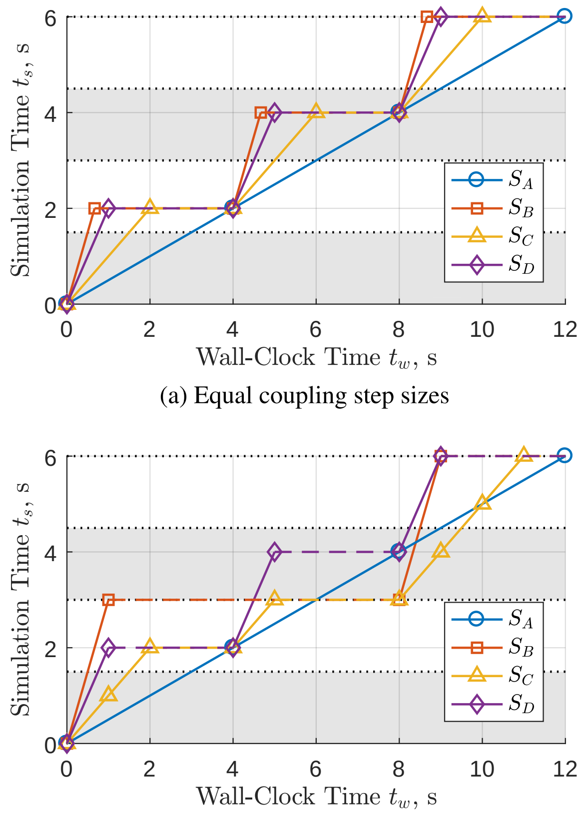
$$(n-1)h_i < \zeta H \leq nh_i < (\zeta+1)H \tag{2a}$$
$$(m-1)h_j < \zeta H \leq mh_j < (\zeta+1)H, \tag{2b}$$

where h_i and h_j represent the coupling step sizes of the FMUs S_i and S_j. A synchronization between two FMUs S_i and S_j takes place, if both FMUs end their coupling step nh_i and mh_j within the same synchronization interval $[\zeta H, (\zeta+1)H)$. In addition to that, it is mandatory, that the previous coupling step $(n-1)h_i$ and $(m-1)h_j$ is outside of the synchronization interval. This avoids multiple data exchange within a synchronization interval. However, the condition in (2) can lead to fewer synchronisations between the subsystems for small synchronisation step sizes, if the individual subsystems end their coupling steps at different synchronisation intervals.

Figure 2 shows the FMU execution and synchronization of a co-simulation example with four FMUs S_A, S_B, S_C and S_D. The FMUs have different coupling step sizes which are assumed with $h_a = 2\,s$, $h_b = 3\,s$, $h_c = 1\,s$ and $h_d = 2\,s$. The synchronization step size is set to $H = 1.4\,s$. The axes show the simulation progress of the individual FMUs with their coupling steps illustrated as cycles. The vertical dashed lines depict the synchronization step sizes. The synchronization and thus the data exchange between the FMUs is illustrated with dark grey solid lines. The first synchronization is in the synchronization interval $[H, 2H)$ at $t_s = 2\,s$. FMUs S_A, S_C and S_D end at the same time and exchange their data. The data exchange is done independently from the simulation progress of FMU S_B. The first synchronization (and data exchange between) all FMUs (including FMU S_B) occurs in the interval $[2H, 3H)$. In the following interval $[3H, 4H)$ only FMU S_C finishes its step. Since no other FMU ends the simulation step in this interval, there is no synchronization and FMU S_C continues the calculation without data exchange. This strategy will be continued for the further intervals. Depending on the coupling step of the FMU, synchronize all FMUs or only parts of the FMUs with each other.

2.1 Synchronization Interval

The synchronization step size H has a direct impact on the synchronization and data exchange of the individual FMUs and thus on their execution behaviour. Figure 3 shows the execution behaviour in terms of the simulation time t_s over the wall-clock time t_w of the co-simulation example in Figure 1 with the FMUs S_A, S_B, S_C and S_D. The solid lines show the execution of the FMUs with a marker at the beginning and the end of the coupling step. The



(a) Equal coupling step sizes

(b) Different coupling step sizes

Figure 3. Simulation time of the FMUs depending on the wall clock time with parallel fast scheduling: (a) Equal coupling step sizes; (b) Different coupling step sizes.

horizontal (dashed) lines between the markers indicate the waiting of the FMU due to the synchronization condition in 2. The synchronization step size is defined as $H = 1.5\,s$. The real-time factor of the FMUs is assumed with $d_a = 2$, $d_b = 0.333$, $d_c = 1$ and $d_d = 0.5$, i.e., for instance FMU S_A with a real-time factor $d_a = 2$ needs $4\,s$ to execute a coupling step of $h_a = 2\,s$. In Figure 3 a all FMUs have the same coupling step sizes $h_a = h_b = h_c = h_d = 2\,s$. Consequently, all FMUs are synchronized at the same time. The FMUs start their execution simultaneously. Due to the different real-time factors, the FMUs end their coupling step at different wall-clock times t_w. For instance, FMU S_C has finished the first coupling step at $t_w = 2\,s$, while FMU S_A ends the first step at $t_w = 4\,s$. After all FMUs have completed their coupling step, data is exchanged. Due to the equal step sizes and thus the equal simulation progress, all FMUs start their next coupling step again simultaneously.

The simulation time t_s over the wall-clock time t_w for different coupling step sizes is shown in Figure 3 b for assumed coupling step sizes $h_a = 2\,s$, $h_b = 3\,s$, $h_c = 1\,s$ and $h_d = 2\,s$. In the first coupling step all FMUs are executed in parallel. FMU S_C is the only FMU that finishes its first step in the synchronization interval $[0\,s, 1.5\,s)$. This means that no synchronisation is necessary and the FMU immediately executes the next coupling step. The first data exchange takes place between the FMUs S_A, S_C and S_D in the synchronization interval $[1.5\,s, 3\,s)$. The three FMUs wait for each other before starting the next coupling step after the data exchange. Synchronization with FMU S_B first occurs in interval $[3\,s, 4.5\,s)$. All FMUs complete their coupling step within this interval. Data is exchanged between all FMUs, despite the different simulation progresses. In the following interval $[4.5\,s, 6\,s)$, only FMU S_C fulfils the synchronization condition, which leads to no synchronization. As a result, FMU S_C executes three coupling steps in a row without exchanging data with the other FMUs. However, at $t_s = 6\,s$, i.e., within interval $[6\,s, 7.5\,s)$, all FMUs end their coupling step and synchronization is performed.

In both diagrams in Figure 3, the simulation duration or the wall-clock time $t_w = 12\,s$ for a simulation time $t_s = 6\,s$. That means that the slowest FMU, i.e., FMU S_A, dominates the simulation duration in both cases. Hence, the synchronization between the FMUs has no effect on the simulation duration in the shown example.

However, the synchronization step size H has an impact on the data exchange between the FMUs and thus on the simulation extrapolation behaviour. Figure 4 shows the extrapolation error E and the real-time factor D of the co-simulation example over the synchronization step size H. Thereby, the extrapolation error E and the real-time factor D of the entire co-simulation were determined from several simulation runs with different synchronization step sizes. The extrapolation error E results from the mean local extrapolation error of all inputs of the FMUs, i.e., it is a measure of the actual extrapolation errors of the entire co-simulation. To identify the impact of the synchronization step size H on the simulation duration and real-time factor D, it is assumed, that all FMUs have the identical real-time factor $d_a = d_b = d_c = d_d = 1$.

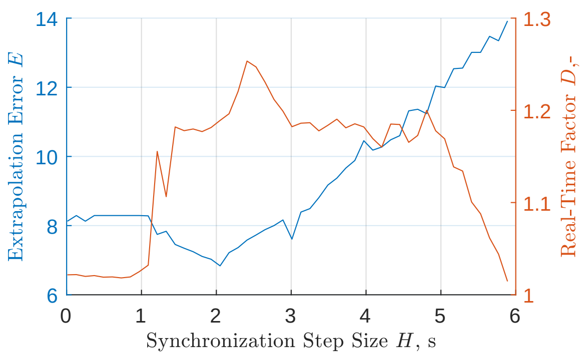


Figure 4. Real-time factor D and extrapolation error $\hat{E}$ of the actual co-simulation depending on the synchronization step size H of the parallel fast scheduling.

For small synchronization step sizes H, the real-time factor of the entire co-simulation is about $D = 1$. After that, the real-time factor increases to about $D = 1.2$, because of the larger synchronization interval, there may be waiting time between FMUs due to different simulation progress, which increases the simulation time of the whole co-simulation. With a synchronization step size of $H = 6\,s$, the real-time factor drops again to $D = 1$ in this example. The synchronization step size $H = 6\,s$ corresponds to the common multiple of all coupling step sizes, i.e., all FMUs end at the same simulation time t_s, which minimize the synchronization time between the FMUs.

The extrapolation error E is almost constant for small synchronization step sizes H in this example. From a synchronization step size of $H = 1\,s$, the extrapolation error E decreases and afterwards increases almost linearly from $H = 2\,s$. Larger synchronization intervals lead to less data exchange between FMUs. This typically means larger extrapolation errors and thus a decrease in simulation accuracy. The lowest extrapolation error and thus the highest accuracy is at $H = 2\,s$. Due to the synchronization condition in (2) smaller synchronization step sizes H do not necessarily lead to better results. For small step sizes, the coupling steps of the FMUs must be finished close to each other in order to be synchronized.

3 Coupling Assessment

The extrapolation error E and the real-time factor D in Figure 4 result from the execution behaviour of the parallel fast scheduling. In addition to the synchronization step size H this depends on the topology of the co-simulation, the coupling step sizes h_i of the individual FMUs and their timing behaviour d_i. Based on these parameters, an extrapolation measure $\hat{E}$ is introduced, which is called extrapolation error estimation. In addition to that the real-time factor $\hat{D}$ of the parallel fast scheduling for the entire co-simulation is estimated.

 DOI
10.3384/ecp21181649

3.1 Extrapolation Assessment

The scheduling defines the synchronization depending on the connections between the FMUs, their coupling step step sizes and the synchronization step size. The synchronization results in the extrapolation behaviour of the coupling signals. In order to derive the extrapolation in the form of an extrapolation error estimation $\hat{E}$, synchronization of the FMUs must be considered. In this context, the interactions between the FMUs are essential. These interactions are defined by the coupling signals, i.e., the connection from an output y_i of FMU S_i to an input u_j of FMU S_j. For the entire co-simulation the connections can be described with the linking matrix:

$$\mathbf{u} = \mathbf{L} \cdot \mathbf{y}, \tag{3}$$

with the vector of all concatenated outputs $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ and the vector of all concatenated inputs $\mathbf{u} = [u_1, u_2, \dots, u_N]$. Without loss of generality, we assume that both vectors have the length N, where N represents the number of connections, i.e., an output can only be connected to one input[2]. Consequently, the linking matrix $\mathbf{L}$ is an orthogonal matrix with the dimension $N \times N$. The connections of the example in Figure 1 can be written using a linking matrix:

$$\begin{bmatrix} u_{ad} \\ u_{ba} \\ u_{cb} \\ u_{cd} \\ u_{dc} \\ u_{da} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} y_{ab} \\ y_{ad} \\ y_{bc} \\ y_{cd} \\ y_{da} \\ y_{dc} \end{bmatrix} \tag{4}$$

However, the linking matrix does not contain any information regarding the dependencies between the FMUs. In order to derive the dependency of the FMUs from the linking matrix, which describes the mapping of the inputs to the outputs, two further matrices $\mathbf{S}$ and $\mathbf{T}$ are introduced. The matrix $\mathbf{S}$ describes the relation of the outputs y_i with the corresponding FMUs S_i and the matrix $\mathbf{T}$ describes relation of the inputs u_j with the corresponding FMUs S_j. Based on the example, the matrices are as follows:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The two matrices $\mathbf{S}$ and $\mathbf{T}$ have the dimension $N \times M$, where M is the number of involved FMUs. In combination with the linking matrix, the dependency matrix $\mathbf{A}$, i.e., the dependencies between the FMUs, is

[2] In case of multiple connected outputs, the outputs can be duplicated.

$$\mathbf{A} = \left(\mathbf{T}^T \cdot \mathbf{L} \cdot \mathbf{S}\right)^T. \tag{6}$$

The dependency matrix $\mathbf{A}$ has the dimension $M \times M$. For the example in Figure 1, the dependency matrix can be written as follows:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \tag{7}$$

where the individual columns and rows are assigned to the FMUs S_A, S_B, S_C and S_D. The entries in the columns indicates the inputs to the FMUs and the rows the outputs. For instance, the first column, i.e., FMU S_A, represents the inputs of FMU S_A, which is connected to FMU S_D. The first row describes the two outputs of FMU S_A to the FMU S_B and S_D.

The dependency matrix $\mathbf{A}$ serves as the basis to determine the extrapolation error estimation $\hat{E}$. In a parallel coupling, where all FMUs S_i have the same coupling step sizes h_i, all inputs u_i are always extrapolated. With respect to the dependency matrix, the extrapolation error estimation can be stated as

$$\hat{E} = \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbf{A}_{ij}. \tag{8}$$

With the definition of the dependency matrix in (6) the extrapolation error estimation $\hat{E}$ corresponds to the number of extrapolated inputs, which implies the assumption that all coupling signals have the same influence on the coupling error and thus on simulation results. Typically, there are coupling signals that have more influence on simulation results than others. This can be considered by extending the dependency matrix $\mathbf{A}$ in (6) with a weighting matrix $\mathbf{C}$ as follows:

$$\mathbf{A} = \left(\mathbf{T}^T \cdot \mathbf{C} \cdot \mathbf{L} \cdot \mathbf{S}\right)^T, \tag{9}$$

where the diagonal matrix $\mathbf{C} = diag(c_1, c_2, ..., c_n)$ has the dimension $N \times N$. The coefficient c_i represents the weighting of the coupling signals. A high weighting factor c_i shows a big impact of the coupling signal on the simulation results and a small weighting factor c_i indicates a little impact on the simulation results. Different methods to weight the coupling signals and their impact on the dependency matrix were discussed by the authors in F. Holzinger and Benedikt (2019) and F. R. Holzinger, Benedikt, and Watzenig (2021).

However, if different coupling step sizes are used, the extrapolation between FMUs and thus the extrapolation error E changes. Consequently, the extrapolation error estimation $\hat{E}$ in (8) is no longer valid, due to the fact that neither the coupling step sizes nor the scheduling is considered. The synchronization and thus the extrapolation of the FMUs depends on the simulation progress nh_i of the individual FMUs S_i according to the synchronization condition in (2).

In the case of a synchronization between two FMUs S_i and S_j, the synchronization step $\Delta\tau_{ij}^{(syn)}$ of FMU S_i is given by the actual simulation progress nh_i and previous simulation progress of the last synchronization $\tilde{n}h_i$:

$$\Delta\tau_{ij}^{(syn)} = nh_i - \tilde{n}h_i, \tag{10}$$

where $\tilde{n}$ indicates the last synchronization between the FMUs S_i and S_j. Due to the different coupling step sizes, synchronization between the FMUs does not take place at every coupling step nh_i. Furthermore, it is not mandatory that the FMUs have the same simulation progress during synchronization, they simply have to be in the same synchronization interval. The different simulation progresses during synchronization have an effect on the extrapolation. If a FMU is more progressed, its results can be used directly and do not have to be extrapolated by the other FMUs, i.e., the inputs can be interpolated. The interpolated part results from the simulation progress of the two coupled FMUs:

$$\Delta\tau_{ij}^{(int)} = mh_j - nh_i \quad \text{for} \quad mh_j > nh_i. \tag{11}$$

Considering FMU S_i, interpolation applies if the simulation progress mh_j of the coupled system S_j is greater than the simulation progress nh_i during the synchronization, otherwise the entire synchronization step for FMU S_i must be extrapolated. Therefore, the interpolated part can be generally written as follows:

$$\Delta\tau_{ij}^{(int)} = \max(0, mh_j - nh_i) \tag{12}$$

the extrapolated part $\Delta\tau_{ij}^{(ext)}$ of the synchronization results from the synchronization step in (10) minus the interpolated part in (12). The interpolated part cannot be larger than the synchronization step, which leads to the following:

$$\begin{aligned} \Delta\tau_{ij}^{(ext)} &= \Delta\tau_{ij}^{(syn)} - \Delta\tau_{ij}^{(int)} \\ &= nh_i - \tilde{n}h_i - \max(0, mh_j - nh_i) \end{aligned} \tag{13}$$

The extrapolated part $\Delta\tau_{ij}^{(ext)}$ in (13) is the time horizon, which is extrapolated within a synchronization step $\Delta\tau_{ij}^{(syn)}$. However, multiple coupling steps can be executed within one synchronization step and the number may vary from synchronization step to synchronization step. For instance, in the first synchronization step of FMU S_C in Figure 2, two coupling steps are executed, whereby in the next synchronization step only one coupling step is executed. The interpretation of the extrapolation error with the dependency matrix $\mathbf{A}$ in (9) is referred to a single coupling step. Due to lack of additional information about the coupling behaviour, it is assumed that the extrapolation impact increases linearly with the number of coupling steps within a synchronization step. That means, if there are multiple coupling steps within a synchronization step, the extrapolated part must be scaled by the impact of multiple coupling steps. For example, three coupling steps are executed within one synchronization step, i.e., $n - \tilde{n} = 3$. The first coupling step starts directly after the last synchronization, i.e., the input data has to be extrapolated only one coupling step size, so there is no scaling for the first coupling step needed. The second coupling step uses the same data for the extrapolation. The extrapolation horizon is two coupling steps in this case. Therefore, the second input value is scaled with the factor 2. In the third coupling step, the extrapolation horizon is already three coupling steps, i.e., scaling with a factor of three. The scaling factor f_i of the synchronization step can be written for the number of $n - \tilde{n}$ coupling steps as follows:

$$f_i = \frac{1}{n-\tilde{n}} \sum_{j=1}^{n-\tilde{n}_i} j = \frac{n-\tilde{n}+1}{2} \tag{14}$$

However, the synchronization time between two FMUs and so the number of coupling steps can change from one synchronization step to the other. Therefore, the extrapolation must be reassessed for each synchronization. In order to derive an overall assessment of the extrapolation behaviour of the coupling signals between two FMUs, it is not sufficient to consider only one synchronization step. The synchronization behaviour and thus the extrapolation behaviour repeats after a certain time. This time corresponds to the least common step size $\bar{H}$ of the coupling step sizes h_i and h_j of the two FMUs and the synchronization step size H. For this time horizon $\bar{H}$, the mean extrapolation error estimation $\hat{E}$ can be calculated for a single coupling signal from FMU S_i to FMU S_j. This normalised extrapolation error estimation or extrapolation ratio $Q(h_i, h_j, H)$ describes the impact of the extrapolation of a coupling signal due to the synchronization step size H and the coupling step sizes h_i and h_j and can be written as:

$$Q(h_i, h_j, H) = \frac{1}{\bar{H}} \sum_{\zeta=1}^{\bar{H}/H} \sum_{n=1}^{\bar{H}/h_i} \sum_{m=1}^{\bar{H}/h_j} s_{e,i} s_{e,j} f_i \Delta\tau_{ij}^{(exp)}, \tag{15}$$

where f_i is the scaling factor regarding the synchronization step in (14), $s_{e,i}$ is the synchronization coefficient for FMU S_i and $s_{e,j}$ is the synchronization coefficient for FMU S_j given as:

$$s_{e,i} = \begin{cases} 1 & \text{if} \quad (n-1)h_i < \zeta H \leq nh_i < (\zeta+1)H \\ 0 & \text{otherwise and} \end{cases} \tag{16a}$$

$$s_{e,j} = \begin{cases} 1 & \text{if} \quad (m-1)h_j < \zeta H \leq mh_j < (\zeta+1)H \\ 0 & \text{otherwise.} \end{cases} \tag{16b}$$

DOI 10.3384/ecp21181649

The coefficient $s_{e,i} = 1$ if the condition in (2a) for FMU S_i is fulfilled. Analogously, the coefficient $s_{e,j} = 1$ if the synchronization condition in (2b) for FMU S_j is fulfilled.

The extrapolation ratio $Q(h_i, h_j, H)$ describes the effect of the parallel fast scheduling on the dependency matrix $\mathbf{A}$ and thus on the extrapolation error estimation $\hat{E}$. The weights of the extrapolation ratio $Q(h_i, h_j, H)$ for the individual step sizes $[h_1, h_2, ..h_M]$ of the individual FMUs can be put together in weighting matrix as follows:

$$\mathbf{W} = \begin{bmatrix} Q(h_1,h_1,H) & \cdots & Q(h_1,h_M,H) \\ \vdots & \ddots & \vdots \\ Q(h_M,h_1,H) & \cdots & Q(h_M,h_M,H) \end{bmatrix} \quad (17)$$

The weighting matrix $\mathbf{W}$ can be element-wise multiplied with the dependency matrix $\mathbf{A}$ to scale it according to the influence of the synchronization of the parallel fast algorithm. Whereby the extrapolation error estimation $\hat{E}$, i.e., the assessment of the extrapolation to the co-simulation, can be determined as follows:

$$\hat{E} = \sum_{i=1}^{M} \sum_{j=1}^{M} \bar{\mathbf{A}}_{ij} = \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbf{W}_{ij} \mathbf{A}_{ij}. \quad (18)$$

3.2 Simulation Duration Assessment

The time of synchronization is defined by the synchronization step size H and the coupling step sizes h_i of the FMUs S_i. The actual duration (relative to wall-clock time) of the synchronization step depends on the duration of the execution of a coupling step nh_i, i.e., of the real-time factor d_i of the FMUs S_i, and the time the FMUs have to wait for each other during synchronization. Therefore, a coupling step implies mainly two parts: the execution of the FMU, dominated by the real-time factor d_i of the FMUs S_i and the synchronization between the FMUs, i.e., the waiting time. During this time, the synchronizing FMUs wait until all have completed their coupling step. The wall-clock time $t_{w,i}$ of the individual FMUs, i.e., the time that is actually needed to execute a coupling step, can be formulated depending on their simulation progress t_s as follows:

$$t_{w,i}(t_s) = \max\left(t_{w,i}(t_s), t_{w,j}(t_s) s_{e,i} s_{e,j}\right) \qquad \forall i, j \in S \quad (19)$$

The actual wall-clock time $t_{w,i}$ of a FMU S_i is determined as the maximum wall-clock time $t_{w,j}$ of all synchronized FMUs S_j. The synchronized FMUs are specified with the coefficients $s_{e,i}$ and $s_{e,j}$ regarding to the synchronization condition in (16). The simulation time can be expressed as $t_s = k\bar{h}$, where $k \in \mathbb{N}$ and $\bar{h}$ is the great-common-divisor step size of the coupling step sizes h_i and the synchronization step size H and consequently the determination of the wall-clock-time in (19) can be rewritten as follows:

$$t_{w,i}(k\bar{h}) = \max\left(t_{w,i}(k\bar{h}), t_{w,j}(k\bar{h}) s_{e,i} s_{e,j}\right) \qquad \forall i, j \in S. \quad (20)$$

Depending on the definition of the simulation time $t_s = k\bar{h}$, the index n of the actual coupling step of a FMU S_i can be determined as $n = \left\lceil \frac{t_s}{h_i} \right\rceil = \left\lceil \frac{k\bar{h}}{h_i} \right\rceil$. Similarly, the index of the synchronization step is $\zeta = \left\lfloor \frac{t_s}{H} \right\rfloor = \left\lfloor \frac{k\bar{h}}{H} \right\rfloor$. Applied to (2a) the synchronization condition results in

$$\left(\left\lceil \frac{k\bar{h}}{h_i} \right\rceil - 1\right) h_i < \left\lfloor \frac{k\bar{h}}{H} \right\rfloor H \leq \left\lceil \frac{k\bar{h}}{h_i} \right\rceil h_i < \left(\left\lfloor \frac{k\bar{h}}{H} \right\rfloor + 1\right) H.$$

The wall-clock time $t_{w,i}$ of a FMUs S_i is individually updated with each execution by its real-time factor d_i. This can be generally written as follows:

$$t_{w,i}((k+1)\bar{h}) = t_{w,i}(k\bar{h}) + d_i h_i s_{d,i} \qquad \forall i \in S, \quad (21)$$

where $s_{d,i}$ indicates the end of a coupling step and can be stated as:

$$s_{d,i} = \begin{cases} 1 & \text{if} \quad \bmod(k\bar{h}, h_i) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

If a coupling step is completed, i.e., the simulation time t_s or $k\bar{h}$ is a multiple of the coupling step size h_i, the coefficient $s_{d,i} = 1$ and its required computation effort $d_i h_i$ is added to the current wall-clock time $t_{w,i}$ of FMU S_i.

In order to approximate the real-time factor of the entire co-simulation it is not sufficient to consider only one synchronization step. As shown in Figure 2, the synchronization between FMUs and so the computational effort changes depending on the simulation progress. However, the synchronization pattern and so the timing repeats after a certain time. This time depends on the coupling step sizes h_i and the synchronization step size H and can be determined by the least-common-multiple step size $\bar{H}$ of all step sizes. That means, to approximate the real-time factor $\hat{D}$ of the entire co-simulation, at least the simulation time interval $t_s = [0, \bar{H}]$ has to be considered.

Finally the estimated real-time factor $\hat{D}$ for the parallel fast scheduling can be determined as

$$\hat{D} = \frac{1}{\bar{H}} \max_{i \in S}(t_{w,i}). \quad (23)$$

Using the extrapolation error estimation $\hat{E}$ in (18) and the real-time factor estimation $\hat{D}$ in (23), the impact of the synchronization step size H on the extrapolation error and the simulation duration can be estimated without costly simulations. The comparison of the extrapolation error estimation $\hat{E}$ and the real-time factor estimation $\hat{D}$ with the actual extrapolation error E and the real-time factor D depending on the synchronization step size H is shown in Figure 5. Both, the extrapolation error estimation $\hat{E}$ and the estimation of the real time factor $\hat{D}$ of the entire co-simulation, show a similar behaviour than the actual extrapolation error E and the actual real-time factor D depending on the synchronization step size H.

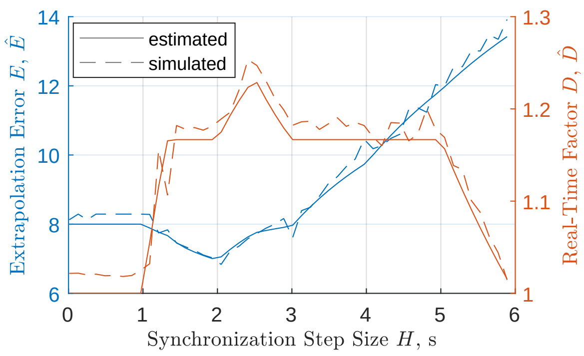


Figure 5. Real-time factor estimation $\hat{D}$ and extrapolation error estimation $\hat{E}$ compared to the actual real-time factor D and extrapolation error E depending on the synchronization step size H of the parallel fast scheduling.

4 Optimal Synchronization Step Size

The extrapolation error E changes strongly depending on the synchronization step size H, see Figure 5. Beyond a certain synchronization step size H, the extrapolation error increases almost linearly with the synchronization step size. This can be mainly explained by the large synchronization intervals that lead to less synchronizations, i.e., less data exchanges between the FMUs, and thus to higher coupling errors. However, the real-time factor D of the entire co-simulation changes only slightly with the synchronization step size H. The effect on the real-time factor and thus on the simulation duration becomes even smaller as soon as the FMUs have different real-time factors d_i. Figure 6 shows the impact of the synchronization step size H on the estimated real-time factor $\hat{D}$ for different real-time factors d_a of FMU S_A, whereby the other FMUs remain with a real-time factor $d_b = d_c = d_d = 1$.

With an increasing real-time factor d_a, the impact of the synchronization step size on the overall real-time factor $\hat{D}$ decreases. At a difference of around 60% of the real-time factor, i.e., $d_a = 1.6$, there is no significant impact of the synchronization time on the overall real-time factor D of the co-simulation. The slowest FMU, i.e., S_A, dominates

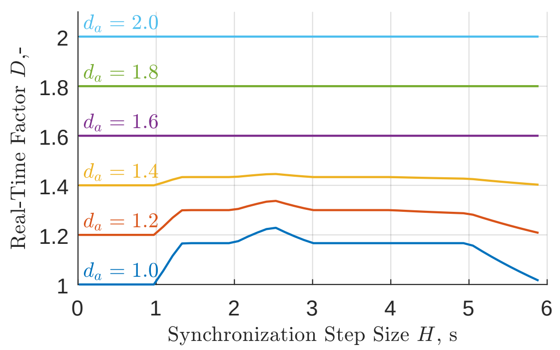


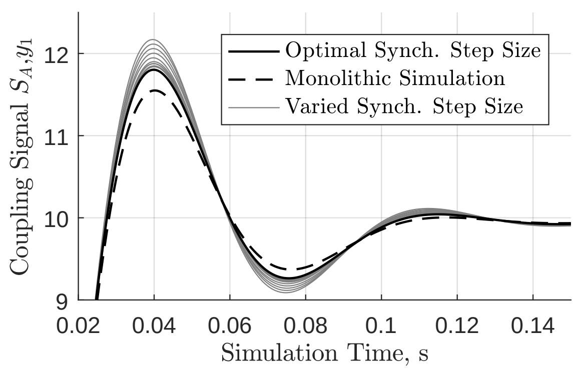
Figure 6. Real-time factor estimation depending on the synchronization step size for different real-time factors of FMU S_A.

the entire timing behaviour, independent on the synchronization step size H.

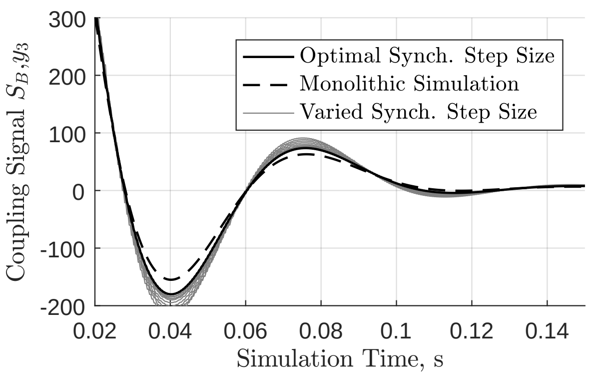
Due to the small effect of the synchronization step size on the timing behaviour and thus on the overall real-time factor $\hat{D}$, the simulation duration is negligible for the determination of the synchronization step size. This means only the extrapolation error estimation $\hat{E}$ is used to find an appropriate synchronization step size. Thus, an optimization problem based on the extrapolation error estimation $\hat{E}$ in (18) can be written as follows:

$$\min_{H} \{\hat{E}\}. \tag{24}$$

A synchronisation step size H is desired that minimizes the extrapolation error estimation $\hat{E}$ in (18) for the considered co-simulation example. In the following, the optimization approach in (24) is applied to the co-simulation example in Figure 1. The coupling step sizes of the FMUs are given with $h_a = 0.2\,ms$, $h_b = 0.3\,ms$, $h_c = 0.1\,ms$ and $h_d = 0.2\,ms$. A detailed description of the FMUs can be found in Benedikt and Drenth (2019) and F. Holzinger and Benedikt (2019). To identify the optimal synchronization step size H, the minimum of the extrapolation error estimation has to be determined. Therefore 20 synchronization steps sizes were evaluated between $H = 0.05\,ms$ and



(a) Simulation result S_A,y_{ab}



(b) Simulation result S_B,y_{bc}

Figure 7. Simulation results for varied synchronization step sizes H, monolithic simulation and optimal synchronization step size H^*: (a) S_A,y_{ab}; (b) S_B,y_{bc}.

DOI 10.3384/ecp21181649

$H = 6\,ms$. The optimal synchronization step size based on the extrapolation error estimation $\hat{E}$ is $H^* = 2\,ms$.

Figure 7 shows the simulation results of the coupling signals y_{ab} and y_{bc} depending on the different synchronization step sizes. The black solid line depicts the simulation result of the optimal synchronization step size H^*. The grey lines show the results of the other evaluated synchronization step sizes. The monolithic simulation, i.e., all subsystems are solved within one model with one single solver without extrapolation, is shown as black dashed line and serves as reference signal. The results with the optimal synchronization step size H^* show the smallest deviation from the reference solution for the coupling signal y_{ab} of FMU S_A and coupling signal y_{bc} for FMU S_B.

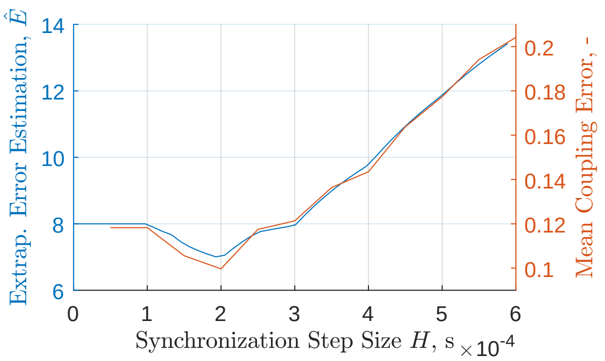


Figure 8. Extrapolation error estimation $\hat{E}$ compared to the mean coupling error w.r.t. the monolithic simulation over synchronization step size H.

The comparison of the determined extrapolation error estimation $\hat{E}$ with the mean coupling error is shown in Figure 8. The mean coupling error corresponds the root mean square error of the simulation results to the reference solution (monolithic simulation) of all coupling signals. The behaviour of the extrapolation error estimation $\hat{E}$ shows similar behaviour to the actual coupling error with respect to the monolithic simulation that occurs. This means that a determination of the optimal synchronization step size H^* using the extrapolation error estimation $\hat{E}$ leads to the optimal solution in terms of the synchronization step size.

5 Conclusion

The co-simulation of FMUs with different coupling step sizes requires suitable synchronization methods to ensure appropriate data exchange of the coupling signals between the FMUs. The presented parallel fast scheduling shows an effective way to synchronize the individual FMUs without unnecessarily extending the simulation duration. The usage of synchronization intervals, where FMUs exchange their data, enables continuous timing behaviour throughout the simulation and is therefore particularly suitable for real-time applications. However, the definition of the synchronization interval has a direct impact on the extrapolation behaviour of the FMUs and thus on the simulation accuracy. Contrary to the expectations, small synchronization intervals do not generally lead to accurate simulation results. Therefore, an optimisation approach was proposed to determine the optimal step size for the synchronization intervals based on an introduced extrapolation measure (extrapolation error estimation). This allows to derive the optimal synchronization step size for the parallel fast scheduling.

The presented determination of the optimal synchronization step size for the parallel fast scheduling based on the estimated extrapolation error shows good results for the example used in this work. However, in future work this approach will be analysed and validated on further examples. The extrapolation assessment depending on the synchronization step size of the co-simulation assumes a ZOH extrapolation for the estimation. The estimation of the extrapolation error for other coupling approaches, e.g., FOH, will be considered in future works.

Acknowledgements

The publication was partly written in cooperation with the VIRTUAL VEHICLE Research GmbH in Graz and partly funded by the COMET K2 – Competence Centers for Excellent Technologies Programme of the Federal Ministry for Transport, Innovation and Technology (bmvit), the Federal Ministry for Digital and Economic Affairs (bmdw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG).

The work presented in this document is co-funded by the EU research and innovation programme Horizon2020 (VISION-xEV: GA No. 824314). The content does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the report lies entirely with the author(s).

References

Benedikt, Martin and Edo Drenth (2019). "Relaxing Stiff System Integration by Smoothing Techniques for Non-iterative Co-simulation". In: *IUTAM Symposium on Solver-Coupling and Co-Simulation*. Ed. by Bernhard Schweizer. Cham: Springer International Publishing, pp. 1–25. ISBN: 978-3-030-14883-6.

Benedikt, Martin, Daniel Watzenig, et al. (2013). "Macro-step-size selection and monitoring of the coupling errof for weak coupled subsystems in the frequency-domain". In: *Proceedings of International Conference on Computational Methods for Coupled Problems in Science and Engineering*, pp. 1–12. ISBN: 978-84-941407-6-1.

Blochwitz, T. et al. (2012). "Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models". In: *Proceedings of the 9th International Modelica Conference*. Munich, Germany: The Modelica Association, pp. 173–184. ISBN: 978-91-7519-826-2. URL: http://dx.doi.org/10.3384/ecp12076173.

Busch, Martin and Bernhard Schweizer (2011). "An explicit approach for controlling the macro-step size of co-simulation methods". In: *Proceedings of the 7th European Nonlinear Dynamics Conference (ENOC 2011): July 24 - 29, 2011,* *Rome, Italy*. Ed. by D. Bernadini, pp. 1–6. ISBN: 978-88-906234-2-4. URL: http://tubiblio.ulb.tu-darmstadt.de/77923/.

Genser, Simon and Martin Benedikt (2018). “A Pre-Step Stabilization Method for Non-Iterative Co-Simulation and Effects of Interface-Jacobians Identification”. In: *Advances in Intelligent Systems and Computing*. ISSN: 2194-5357.

Glumac, Slaven and Zdenko Kovacic (2018). “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. Rome, Italy: ACM, pp. 157–160. ISBN: 978-1-4503-5092-1. DOI: 10.1145/3200921.3200924. URL: http://doi.acm.org/10.1145/3200921.3200924.

Haid, Timo et al. (2018). “A model-based corrector approach for explicit co-simulation using subspace identification”. In: The 5th Joint International Conference on Multibody System Dynamics.

Holzinger, Franz and Martin Benedikt (2019). “Optimal Trigger Sequence for Non-Iterative Co-Simulation:” in: *Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications. Prague, Czech Republic: SCITEPRESS - Science and Technology Publications, pp. 80–87. ISBN: 978-989-758-381-0. DOI: 10.5220/0007833800800087.

Holzinger, Franz Rudolf, Martin Benedikt, and Daniel Watzenig (2021). “Optimal Trigger Sequence for Non-iterative Co-simulation with Different Coupling Step Sizes”. In: *Simulation and Modeling Methodologies, Technologies and Applications*. Ed. by Mohammad S. Obaidat, Tuncer Ören, and Helena Szczerbicka. Cham: Springer International Publishing, pp. 83–103. ISBN: 978-3-030-55867-3.

Kübler, R. and W Schiehlen (2000). “Modular Simulation in Multibody System Dynamics”. In: *Multibody System Dynamics* 4.

Matlab Simulink, R2021a (2021). *Co-Simulation Execution and Numerical Compensation*. (https://www.mathworks.com/help/simulink/ug/co-simulation-execution-and-numerical-compensation.html). Accessed: 2021-04-11.

Model.CONNECT™, R2020a (2020). *AVL’s open model integration and co-simulation platform*. (https://www.avl.com/-/model-connect). Accessed: 2020-02-04.

Oakes, Bentley et al. (2020). “Hint-Based Configuration of Co-simulations with Algebraic Loops”. In: *Simulation and Modeling Methodologies, Technologies and Applications*. Springer International Publishing, pp. 1–28. ISBN: 978-3-030-55866-6. DOI: 10.1007/978-3-030-55867-3_1.

Sadjina, Severin et al. (2016-11). “Energy conservation and power bonds in co-simulations: non-iterative adaptive step size control and error estimation”. In: *Engineering with Computers* 33.3, pp. 607–620. ISSN: 1435-5663. DOI: 10.1007/s00366-016-0492-8. URL: http://dx.doi.org/10.1007/s00366-016-0492-8.