Engineering with Computers
https://doi.org/10.1007/s00366-023-01791-1

ORIGINAL ARTICLE

Transmission line modeling co-simulation with distributed delay-size control using steady-state identification

Robert Braun[1] · Robert Hällqvist[2] · Dag Fritzson[3]

Received: 8 September 2022 / Accepted: 31 January 2023
© The Author(s) 2023

Abstract

Combining performance and numerical stability is a key issue in co-simulation. The Transmission Line Modeling method uses physically motivated communication delays to ensure numerical stability for stiff connections. However, using a fixed communication delay may limit performance for some models. This paper proposes Steady-State Identification for enabling variable communication delays. Three algorithms for online Steady-State Identification are evaluated in three different co-simulation models. All algorithms are able to identify steady state and can thereby determine when communication delays can be allowed to increase without compromising accuracy and stability. The results show a reduction in number of the solver derivative evaluations by roughly 40–60% depending on the model. The proposed method additionally supports connections with asymmetric communication delays, which allows each sub-model to independently control the delay of its input variables. Models supporting delay-size control can thereby be connected to those that do not, so that the step length of each individual sub-model is maximized. Controlling the delay-size in sub-models also makes the method independent of the master co-simulation algorithm.

Keywords Transmission line modeling · Co-simulation · Steady-state identification · Step-size control

1 Introduction

Co-simulation is commonly used for connecting sub-models with independent numerical solvers. This technique may improve modularity and reusability of simulation models, facilitate collaboration between organizations, and enable the use of the most suitable modeling environment and solver algorithm for each part of a larger model. However, numerical instability remains a key issue in co-simulation, especially when connecting stiff mechanical sub-models. Sampled communication between isolated solvers may introduce time delays on interface variables. These may cause instability or inaccuracy if left unhandled. Furthermore, aliasing effects may arise if the sampling rate is too low compared to the time scale of the sub-models [1–3]. A well-known solution to these problems is to use Transmission Line Modeling (TLM) [4–6]. TLM addresses the stability problems by ensuring that the introduced time delays are physically motivated. As a consequence, no non-physical delays need to be introduced, and numerical errors can be completely avoided.

While TLM offers the possibility to effectively eliminate numerical instability, it comes at the cost of enforcing a fixed global maximum communication step-size that is limited by the physical time delay in the elements being modeled. Such a fixed global maximum step-size can negatively impact the simulation performance if a quantified coupling error is unacceptable for accuracy or stability reasons. In most cases, the internal solver integration step-size in a co-simulation unit is not allowed to be larger than the communication step-size. Thus, the steps of the internal solver will also be limited by the physical delays. This can severely limit

Robert Hällqvist and Dag Fritzson have contributed equally to this work.

✉ Robert Braun
robert.braun@liu.se

Robert Hällqvist
robert.hallqvist@liu.se

Dag Fritzson
dag.fritzson@skf.com

1 Division of Fluid and Mechatronic Systems, Linköping University, Linköping, Sweden

2 Saab AB, Linköping, Sweden

3 AB SKF, Göteborg, Sweden

simulation performance, especially if the time scale of a sub-model is significantly larger than the timescale enforced by the TLM connection.

In this paper, we propose the use of distributed adaptive communication delay-size control. When transients are small, it should be possible to relax the maximum step-size limitation and temporarily allow larger delays. Steady-state identification is suggested for controlling delay-size. The purpose of the presented research is to enable numerically stable co-simulation without compromising simulation performance.

1.1 Research questions

The following topics are investigated in the paper. Example models are used for numerical assessments.

1. *Bi-lateral delay lines with asymmetric delays* Traditional TLM connections, also known as *bi-lateral delay lines*, have symmetric numerical properties. As a consequence, the delays have the same magnitude in both directions. However, the proposed approach to distributed delay-size control requires asymmetric delays. It must be ensured that this is feasible and that it does not compromise simulation accuracy.
2. *Matching characteristic impedance* The physical properties of a TLM connection are computed from the combination of the delay and the characteristic impedance. The characteristic impedance Z_c is the ratio between input pressure and input flow of an infinitely long transmission line, see for example Boden et al. [7], and it quantifies the resistance a transmission line presents to changes in pressure. It must be investigated whether or not it is necessary to change the impedance accordingly to match the change in the delays.
3. *Varying delays and impedance during simulation* For the proposed method to be effective, the delays and characteristic impedance of the TLM connection must change online during co-simulation. This must be possible without introducing non-physical numerical delays.
4. *Error criteria for delay-size control* Controlling the size of the delays requires some criteria. Alternatives will be identified and their feasibility investigated.
5. *Evaluation of performance improvements* Finally, if the method is feasible, a numerical assessment of the effects on stability and performance must be conducted.

1.2 Related work

An alternative solution to the proposal presented herein is to invoke an *adaptive communication step-size* [8]. In such a case, the communication step-size for the entire composite model is modified to meet specified tolerances. Adaptive communication step-size can be implemented either using implicit communication schemes [8, 9], explicit schemes [10], or both [11]. In comparison to distributed adaptive delay-size, as described in this paper, an implicit adaptive communication step-size control requires all sub-models to internally support so called solver rollback, that is the capability of the solver to revert to a previously stored stage in the simulation. Hence, the co-simulation will be limited to include only sub-models supporting this feature. Furthermore, with adaptive communication step-size, the communication step will be reduced everywhere, even for parts of the composite model that might not need it, possibly rendering sub-optimal execution times.

Another related technique is *dynamic decoupling* [12, 13]. With dynamic decoupling, models are split into weakly coupled sub-models by identifying nodes with large capacitance, where effort variables change slowly, or large inductance, where flow variables change slowly. This minimizes the error from the communication delay. A drawback is that models can only be decoupled in nodes with slow dynamics, and the transient behaviors of the interface variables are not considered.

A third related approach is to make communication between weakly coupled sub-models more accurate through the use of extrapolation of interchanged variables. Accuracy can be further improved by analyzing the current trend of each variable and adopting the extrapolation method accordingly. Both synchronous approaches [14, 15] and asynchronous algorithms using explicit error estimation [16] exists. Éguillon et al. [10] later presented F_3ORNITS, a co-simulation algorithm providing flexible extrapolation, time-stepping, and scheduling. In comparison with the method proposed in this paper, all of the aforementioned methods rely on weak coupling methods and cannot on their own guarantee numerical stability for a general case model.

Other researchers have previously investigated TLM with adaptive delay-size control. One approach is to use parasitic inductance for error control when simulating fluid systems [17]. There are also similar studies for thermal diffusion [18] and electrical circuits [19]. These studies focused on parallel simulation in a single simulation tool and did not consider co-simulation.

2 Background

2.1 Steady-state identification

SSI techniques can be useful in many different applications ranging from automated model calibration and validation [20] to simulation step-size control. Krus exploited SSI for managing evolving model timescales for optimal simulation performance in [21]. In [22] Hallqvist et al. applied and evaluated four different well-established SSI techniques on aircraft in-situ measurements. The evaluation criteria focused on objectivity and scalability in addition to the SSI functionality. These criteria are also the most relevant for the research presented in this article.

Three methods for online SSI are analyzed here. The first is a *rectangular sliding window* (RSW) with an absolute tolerance. When the difference between the maximum and minimum value of a signal $y(t)$ of interest in the window is below the specified tolerance y_{tol}, then the signal is considered to be at steady state

$$\begin{aligned} \mathrm{SS}(t_i) &= \max(y(t)) - \min(y(t)) < y_{\mathrm{tol}} \\ &\quad \forall t \in [t_i - \Delta t_{\mathrm{w}}, t_i], \end{aligned} \tag{1}$$

where the length of the sliding window is represented by Δt_{w} in Eq. (1). This method requires two parameters to be specified, the window length and the absolute tolerance. Both of these need to be calibrated depending on both the order of magnitude of the signal and the expected oscillation frequency. On the other hand, the required implementation effort is small and the method is easy to understand, which are characteristics that should not be underestimated particularly when considering technology adoption in industry.

A more sophisticated approach is to apply the F-test type statistic known as the *ratio of differently estimated variances* [23], which operates on a sliding window of data just as the technique presented in Eq. (1). The ratio

$$R = \frac{\sigma_1^2}{\sigma_2^2} \tag{2}$$

is equal to one at steady state if there is no random noise present. In the case of SSI for step-size control, there is no random noise enforcing a subjective threshold value on R. In Eq. (2), the variance is estimated in two different ways: as the mean square deviations from the average μ

$$\sigma_1^2(y) = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \mu)^2 \tag{3}$$

and as the mean of squared difference of successive data [24]

$$\sigma_2^2(y) = \frac{1}{n-1}\sum_{i=1}^{n-1}(y_{i+1} - y_i)^2. \tag{4}$$

The current data set in the sliding window is denoted by y in both Eqs. (3) and (4). Furthermore, μ is the average value of all data in the current window, continuously estimated across the n samples in the sliding window.

Cao et al. proposed an improvement to the F-test method by exploiting an exponentially weighted moving average instead of a sliding window when estimating the two variances of Eq. (2) [23]. This improvement, denoted as *R-test*, reduces the overhead associated with the method, but it does introduce a total of three subjective parameters, $[\lambda_1, \lambda_2, \lambda_3]$ that need to be specified. The first variance is computed as the mean square deviation from the filtered data as

$$\sigma_1^2(y) = \frac{2-\lambda_1}{2}v_{f,i}^2 = \frac{2-\lambda_1}{2}\lambda_2\left(y_i - y_{f,i-1}\right)^2 v_{f,i-1}^2, \tag{5}$$

where

$$y_{f,i} = \lambda_1 y_i + \left(1-\lambda_1\right)y_{f,i-1} \tag{6}$$

is the filtered data. The second variance is computed using the mean square differences of successive filtered data, $\delta_{f,i}^2$, as

$$\sigma_2^2(y) = \frac{\delta_{f,i}^2}{2} = \frac{\lambda_3\left(y_i - y_{i-1}\right)^2 + \left(1-\lambda_3\right)\delta_{f,i-1}^2}{2}. \tag{7}$$

If the simulation is assumed to start at steady state, the initial value for $y_{f,0}$ becomes the initial value for the signal and $\delta_{f,0}^2$ becomes zero.

Even though both the F-test and R-test approaches to SSI are developed to be general to any input signal, generality issues arise when applying the techniques on simulation results without noise. Periodic oscillations always generate a non-steady condition if noise is absent. This means that there is no generic tunable parameter to separate irrelevant oscillation amplitudes from relevant ones without modifying the techniques. Measured signals always include appended noise, and the magnitude of this noise is typically related to the magnitude of the measured quantity via the range of the sensor; a relationship that introduces a generic calibration parameter via a tolerance on Eq. (2). One way to mitigate this drawback on noiseless modeled signals could be to include modeled noise; however, the noise connection to the range of the quantity of interest also needs to be modeled for the solution to be general. This method is used in the following experiments.

2.2 Transmission line modeling

The TLM boundary equations can be derived from the wave equation and the telegrapher's equation [25]. They describe the relationship between flow variables (f) and effort variables (e) on each side of a TLM element (Fig. 1).

The effort variables on either side of the element

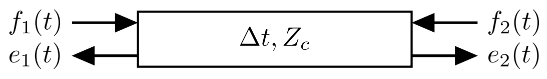

Fig. 1 A TLM element defines a relationship between flow and effort variables on each side

$$e_1(t) = e_2(t - \Delta t) + Z_c\left[f_1(t) + f_2(t - \Delta t)\right]$$
$$e_2(t) = e_1(t - \Delta t) + Z_c\left[f_2(t) + f_1(t - \Delta t)\right] \tag{8}$$

depend on the flow variable on the same side and the flow and effort variables on the other side. Z_c is the characteristic impedance of the element. Note that the TLM element is symmetric; see Eq. (8).

The two main properties of the element are the time delay Δt and the characteristic impedance Z_c. These together yield the capacitance

$$C = \frac{\Delta t}{Z_c} \tag{9}$$

and inductance

$$L = Z_c \Delta t \tag{10}$$

of the element where C and L define the physical behavior depending on the element's domain. In a mechanical element, for example, the capacitance and inductance correspond to the compressibility and inertia, respectively.

The TLM equations enable asynchronous co-simulation [4]. Two interconnected sub-models interact by sending flow variables to the TLM element whenever they are produced. In the TLM element, a co-simulation master algorithm is used to store the flow variables in interpolation tables. Whenever a sub-model requires an effort variable, it can then be interpolated for a particular simulation time. A prerequisite for this to work is that each sub-model must be able to produce output variables at a frequency equal to or larger than the inverse of the TLM time delay. In practice, when working with integrating solvers for systems of differential equations, this means that the internal integration step in the sub-models must be equal to or smaller than Δt. If the communicating sub-models are unable to produce data simultaneously, then the integration step must be limited to $\Delta t/2$ [26].

The time delay in a TLM connection corresponds to an inherent physical property of the modeled element. Hence, no numerical time delays are required. TLM thereby offers co-simulation with guaranteed numerical stability as analytically shown by Braun et al. in [27]. TLM with asynchronous communication using interpolation tables is currently implemented in the OMSimulator master simulation tool [28], which was used for the experiments in this paper.

3 Delay-size control

The limitation on the internal integration steps in the sub-model solvers may induce performance limitations. If the limitation on the integration step can be relaxed, it may be possible to achieve numerical stability without compromising performance. The implications of such a relaxation on the physical properties of the TLM connection are investigated analytically in this section.

As mentioned, the two main properties in a TLM element are inductance and capacitance. The relationships between L, C, e, and f are

$$e_L(t) = L\frac{\mathrm{d}f_L}{\mathrm{d}t} \tag{11}$$

and

$$f_C(t) = C\frac{\mathrm{d}e_C}{\mathrm{d}t}, \tag{12}$$

where the subscripts indicate the origin of the contribution to the variable in question. Hence, L produces an effort proportional to the change rate of the flow, while C produces a flow proportional to the change rate of the effort. At steady state, it holds that

$$\frac{\mathrm{d}f}{\mathrm{d}t} \approx 0 \tag{13}$$

which combined with Eq. (11) shows that the effects of both L and e_L are negligible in steady state.

We now introduce a new factor κ, which is the increase factor for the delay. Both the characteristic impedance and the time delay are then multiplied by κ

$$Z_c' = \kappa Z_c, \tag{14}$$

$$\Delta t' = \kappa \Delta t, \tag{15}$$

where $1 \leq \kappa \leq \kappa_{\max}$. Furthermore, Z_c' and $\Delta t'$ render a new inductance

$$L' = Z_c' \Delta t' = \kappa Z_c \kappa \Delta t = \kappa^2 L \tag{16}$$

and a new capacitance C'

$$C' = \frac{\Delta t'}{Z_c'} = \frac{\kappa \Delta t}{\kappa Z_c} = C. \tag{17}$$

As can be seen, $C' = C$ while L increases by the square of κ. This means that when close to steady state, it should be possible to increase the TLM time delay if it can be compensated for by increasing the characteristic impedance by the same magnitude. The capacitance will not change, and the inductance will not affect the results at steady state.

3.1 Asymmetric communication delays

Because the described delay-size control is conducted within each included sub-model, each sub-model computes its own delay and the corresponding compensated characteristic impedance. As a result, the proposed method must support asymmetric delays, thus enabling delay-size control in a distributed fashion rather than letting the master algorithm control the delays. Additionally, support for asymmetric delays enables sub-models that support delay-size control to be connected to sub-models that do not. This provides a flexibility and modularity similar to the work presented in [10].

3.2 SSI for delay-size control

A TLM connection is here considered to be at steady state if both the effort and flow variables are at steady state intuitively rendering a multivariate SSI problem. As highlighted by Xu et al. [29], multivariate SSI is often dealt with by means of combining all relevant univariate conditions via a logical *and* to identify process overall steady state. Even though more advanced multivariate SSI techniques exist, for example, Xu et al. who expand on the *R-test* by means of forming a ratio between differently estimated co-variance matrices instead of the variances [29], the traditional approach where

$$\mathrm{SS}_{\mathrm{process}} = \prod_{i=a}^{b} SS(i) \tag{18}$$

was used for the purposes of this article.

The obvious approach to identifying steady state in a TLM connection is to equate Eq. (18) for $i = e_{\mathrm{L}}$ and $i = f_{\mathrm{C}}$, as described in Eqs. (11) and (12), exploiting a suitable univariate SSI technique. However, as described earlier in this section, analyzing the physics of the TLM connections renders a second univariate option via Eq. (13).

An SSI algorithm is used here to determine whether the delay-size should be increased or decreased. Figure 2 shows how delay-size control is implemented within a sub-model. After each finished internal solver step, the algorithm checks for steady state. Should steady state not be detected, and the delay is not at the minimum, the delay will be reduced and the solver will roll back and restart with a shorter step. How much the delay is reduced or increased at every step may likely affect simulation performance, but this is not further investigated in this work. In the experiments presented below, a simple method is used where the delay is either halved or doubled until it reaches the maximum or minimum size. This strategy is here deemed as sufficient for the purposes of this prototype implementation; however, future research on the topic could prove valuable for performance improvements. The internal solver contains a callback function for evaluating model state derivatives. Input variables are automatically fetched from within this function, and does therefore not appear in the flowchart.

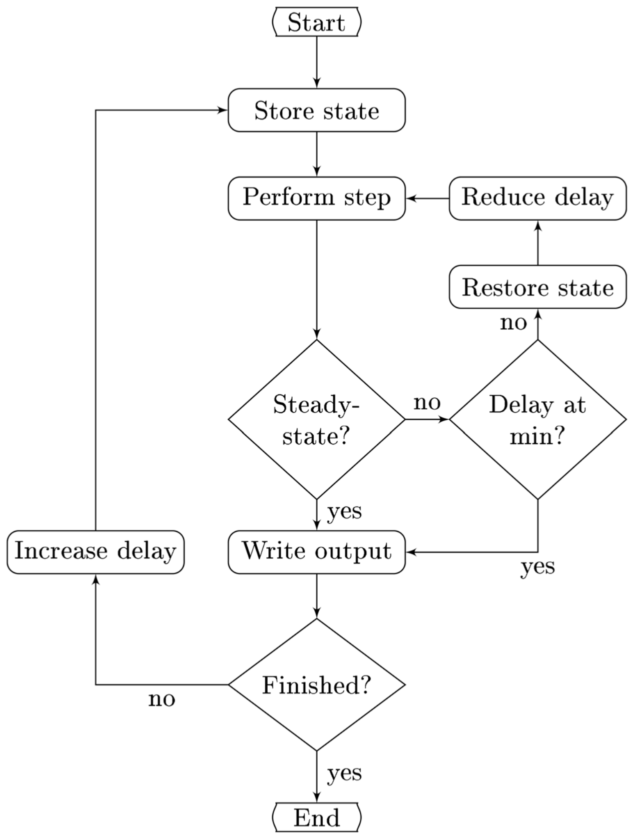

Fig. 2 Flowchart describing the implementation of distributed delay-size control

4 Test models

A total of three test models, consisting of two or more subsystem models coupled using TLM, serve as the foundation for evaluating the technology developed herein.

4.1 Spring-mass model

The first model consists of a 1D spring-mass system with three masses; see Fig. 3. It is divided into two sub-models, where the rightmost contains two of the masses. Thus, the sub-model contains the second-order internal dynamics. After 0.1 s, the leftmost mass is subjected to a step force. The *CVODE* [30] solver is used to solve the system of ordinary differential equations

$$\begin{cases} M\ddot{x}_1 + B\dot{x}_1 = F_{\mathrm{step}} - F_{\mathrm{tlm}}, \\ M\ddot{x}_2 + B\dot{x}_2 + K_1(x_2 - x_3) = F_{\mathrm{tlm}}, \\ M\ddot{x}_3 + B\dot{x}_3 + K_1(x_3 - x_2) + K_2x_3 = 0. \end{cases} \tag{19}$$

Furthermore, the model parameters are listed in Table 1.

Table 1 Parameters used in the spring-mass model

Name	Value	Unit
M	5×10^{-3}	kg
B	0.2	Ns/m
K_1	90	N/m
K_2	140	N/m
F_{step}	100	N

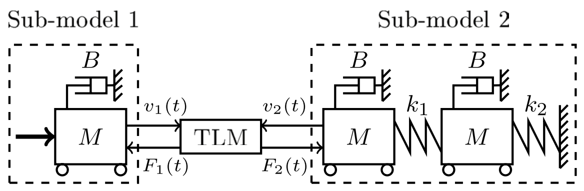

Fig. 3 A test model consisting of a three-mass system divided into two sub-models. The rightmost sub-model has second-order internal dynamics

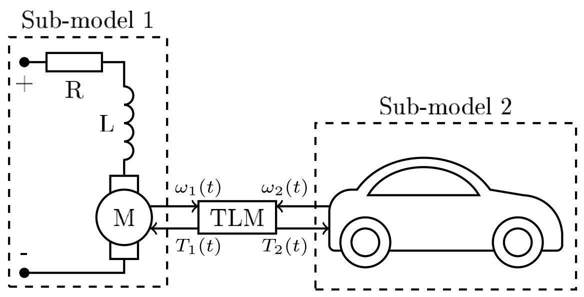

Fig. 4 A model of an electric car powered by a DC motor performing a drive cycle

As reference, the model is simulated without delay-size control.

4.2 Drive cycle model

The second test model is an electric vehicle simulated performing a drive cycle. While the model is quite simple, it still represents a more realistic scenario compared to the first test model. The model is separated into a sub-model of a DC motor and a vehicle sub-model; see Fig. 4.

The relationship between torque (T_{tlm}), velocity (ω_{tlm}), voltage (V), and current (I) in the motor can be expressed as

$$V = IR + \frac{\mathrm{d}I}{\mathrm{d}t}L + K_{\text{e}}\omega_{\text{tlm}} \tag{20}$$

and

$$T_{\text{tlm}} + \frac{\mathrm{d}\omega_{\text{tlm}}}{\mathrm{d}t}J + B\omega_{\text{tlm}} = K_{\text{t}}I, \tag{21}$$

Table 2 Parameters for the electric car test model

Name	Value	Description
K_{e}	7×10^{-5} Vs/rad	Electric constant
K_{t}	7×10^{-5} Nm/A	Torque constant
R	4 Ohm	Motor resistance
L	1×10^{-5} H	Motor inductance
J	0.1	Moment of inertia

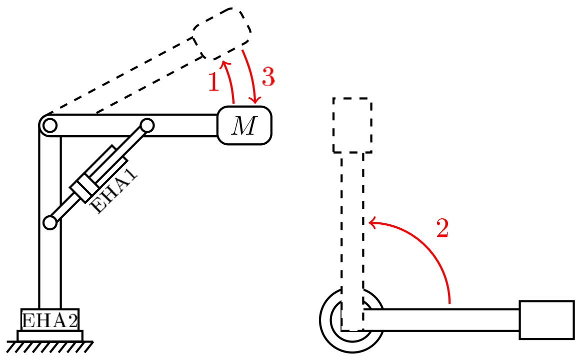

Fig. 5 A hydraulic robot model is connected to two models of electro-hydraulic actuators. The latter models support delay-size control

where the parameters for Eqs. (20) and (21) are provided in Table 2.

The motion of the car is simulated by applying Newton's second law with rolling resistance, air drag, and gravitational pull. Again, the CVODE solver is used to solve the system of ordinary differential equations. The car is simulated performing the New European Drive Cycle (NEDC) together with some variations in road altitude.

4.3 Hydraulic robot model

The third and final model used for the evaluation is a 3-dimensional (3D) hydraulic robot. This model serves two different validation purposes in the presented context. First, it validates the method by incorporating an adaptive delay-size on a 3D mechanical problem. Second, because the model contains connections between sub-models that support adaptive delay-size and those that do not, it is also used to validate the flexibility of the method.

A schematic overview of the robot model is provided in Fig. 5. It performs a three-stage motion

1. Raise the arm by approximately 25°
2. Rotate 90°
3. Lower the arm.

The motion is controlled by two electro-hydraulic actuators (EHAs), as shown in Fig. 6. For the rotating EHA, the

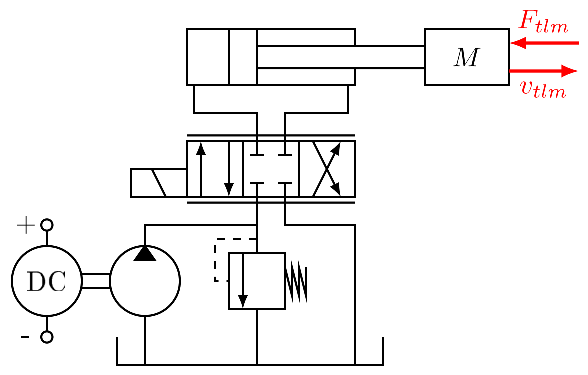

Fig. 6 Flow-controlled electro-hydraulic actuator (EHA) used in the 3D model

hydraulic piston is replaced by a hydraulic motor. An EHA is a self-contained actuator with an electric motor, a hydraulic pump, a hydraulic actuator, and possibly control valves. Such an actuator eliminates the need for a central hydraulic supply system and makes it possible for each actuator to operate independently. Accordingly, the two EHAs are modeled as independent sub-models. The 3D mechanical system is modeled using the *Modelica standard library* [31, 32] and OpenModelica [33]. The latter sub-model does not support adaptive communication delays, whereas the two EHA sub-models do.

5 Experiments

5.1 SSI for delay-size control

The hypothesis that the SSI of a TLM connection is a univariate problem, under the assumption presented in Eq. (13), is tested by comparing univariate SSI to the results from traditional multivariate SSI, here realized by invoking Eq. (18).

5.2 Adaptive delay-size

The suggested method was validated by allowing the maximum delay-size for both interconnected models to increase 20 times when steady state was identified. Each model controlled its own delay-size by identifying steady-state conditions using the RSW method. This relaxed the maximum limitation on the solver integration step-size and thereby the number of derivative evaluations. The delay-size and the number of derivative evaluations were logged to verify the usefulness of the method.

5.3 Asymmetric communication delays

An important property of the proposed method is that it must support asymmetric delays. This makes it possible for each

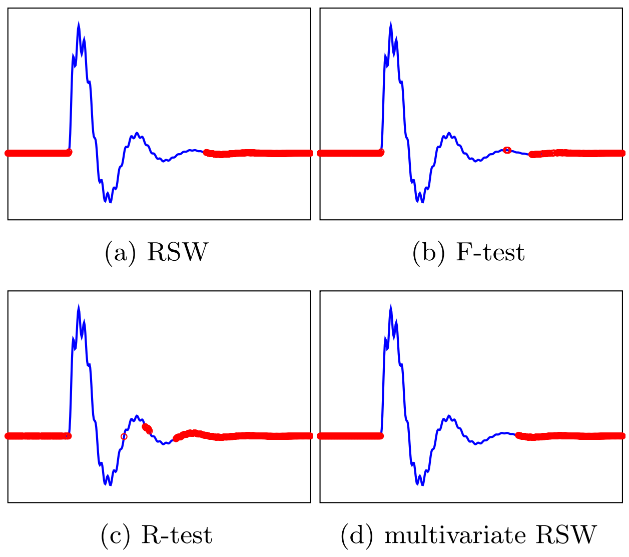

Fig. 7 Results from SSI on spring-mass velocity using the three univariate methods and the multivariate RSW

sub-model to control the delay-size in a distributed fashion rather than letting the master algorithm control the delays. In this way, sub-models supporting delay-size control can be connected to sub-models that do not.

Asymmetric delays were tested by simulating the spring-mass model with $\kappa_1 = 1$ and $\kappa_2 = 10$. Hence, the second sub-model was allowed to increase the delay-size ten times when steady state was observed, while the first sub-model was forced to always use the physically motivated delay. In this way, it could be verified that a sub-model supporting delay-size control can be connected to a regular model using TLM.

6 Results

The results from applying the experiments described in Sect. 5 on the test models, see Sect. 4, are presented in this section.

6.1 SSI for delay-size control

Simulated effort and flow variables from all three test models served as the foundation for comparing the different SSI techniques. The SSI results related to the spring-mass model are shown in Fig. 7, the drive cycle model in Fig. 8, and the hydraulic robot model in Fig. 9.

The RSW univariate and multivariate approaches rendered close to identical steady-state intervals in all investigated experiments, indicating that a univariate approach is sufficient for TLM connections. In other words, the effort variable does not have an impact on the results if the flow variable is in steady state. Several different advantages

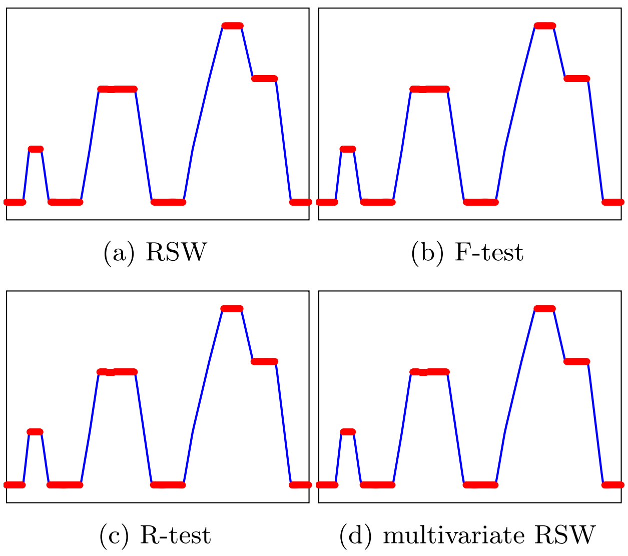

Fig. 8 Results from SSI on angular velocity in the drive cycle model using the three univariate methods and the multivariate RSW

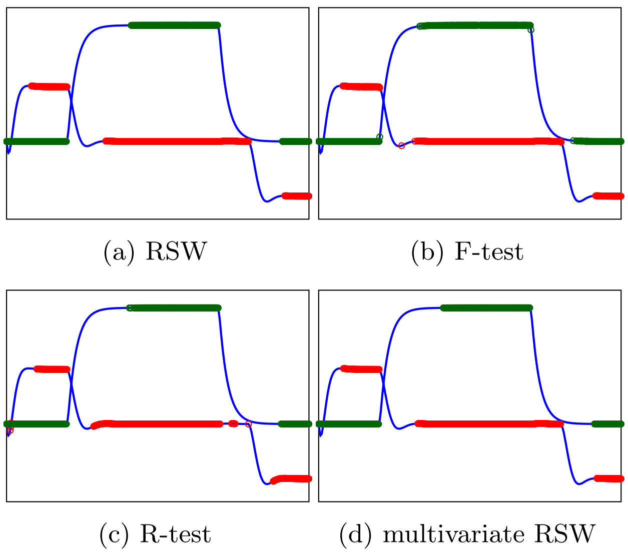

Fig. 9 Results from SSI on piston and motor velocities for the robot model using the three univariate methods and the multivariate RSW

related to generality emerge as a result. All of the investigated univariate SSI methods have drawbacks in terms of generality within the presented context and need to be calibrated to the range of the input signal. The univariate solution renders half the number of calibration parameters compared to multivariate in the case of SSI on a TLM connection.

No further evaluation concerning the selection of which univariate method to employ, in combination with the assumption of Fig. 13, was done here. However, results from all methods are presented to demonstrate that they all

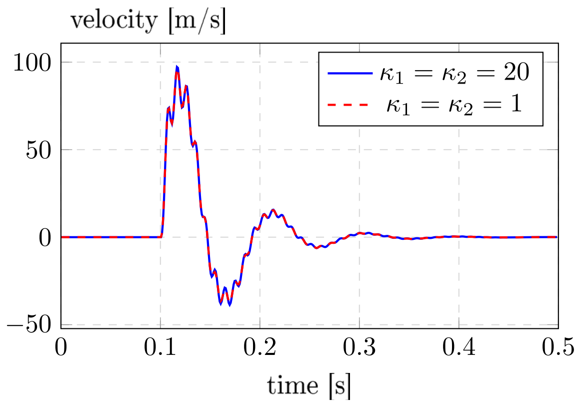

Fig. 10 Velocity in the spring-mass model with and without delay-size control

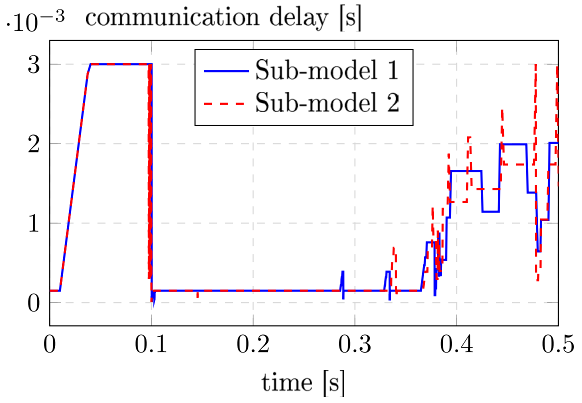

Fig. 11 Communication delays for the spring-mass sub-models with $\kappa_1 = \kappa_2 = 20$

perform similarly if calibrated to do so. Modeled noise was appended to the input signal of the F-test and R-test methods to allow for calibration.

6.2 Adaptive delay-size

Simulation results for the spring-mass model when using adaptive delay-size with $\kappa_1 = \kappa_2 = 20$ are shown in Fig. 10 along with the corresponding results generated without an adaptive delay-size. No significant differences can be observed when comparing the two solutions.

Figure 11 shows the delay-size used by each sub-module when $\kappa = 20$. Both models now use a larger delay-size at the beginning and end of the simulation, where steady-state conditions are identified. Note that the delay-sizes are roughly similar but also contain noticeable differences.

In Fig. 12, the number of derivative evaluations with and without adaptive delay-size is shown. The black line indicates the number of evaluations with the reference

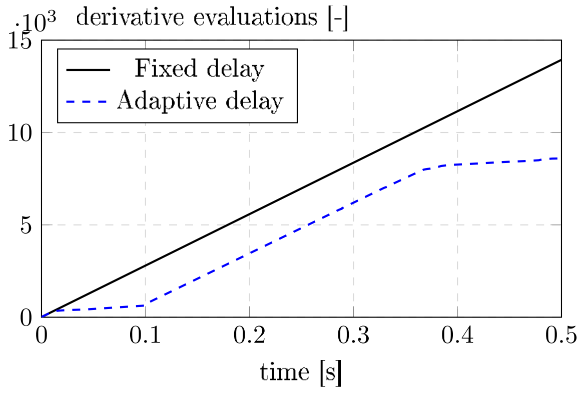

Fig. 12 Total number of derivative evaluations for the spring-mass sub-models with $\kappa_1 = \kappa_2 = 20$. The number of derivative evaluations when a fixed step-size is applied is shown as solid, whereas the evaluations with an implemented variable steps size is shown as dashed

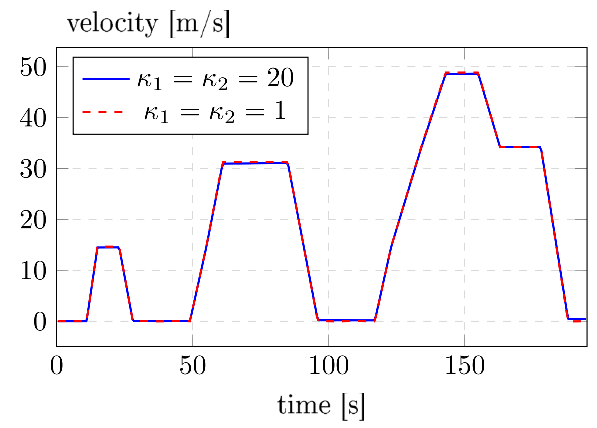

Fig. 13 Vehicle speed in the drive cycle model with and without delay-size control

simulation without delay-size control ($\kappa_1 = \kappa_2 = 1$). The workload for the simulation was reduced from approximately 7000 to 4300 evaluations for each sub-model, a reduction of 38%.

A similar experiment was performed on the drive cycle model. Vehicle speed is shown in Fig. 13, while Fig. 14 shows the delay-size used by each sub-module. Delay-size was increased to the maximum when the car was moving at constant speed. The number of derivative evaluations for the two sub-models is shown in Fig. 15. The black line indicates the number of evaluations with the reference simulation without delay-size control ($\kappa_1 = \kappa_2 = 1$). In total, the simulation requires approximately 58% fewer evaluations with delay-size control compared to without.

Simulation results and communication delay for the robot model are shown in Figs. 16 and 17, respectively. As expected, both EHAs use maximum delays when standing still and a smaller delay when moving.

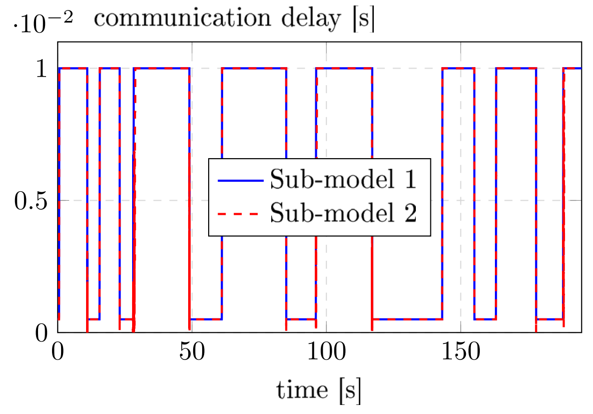

Fig. 14 Communication delay for the car and motor sub-models with $\kappa_1 = \kappa_2 = 20$

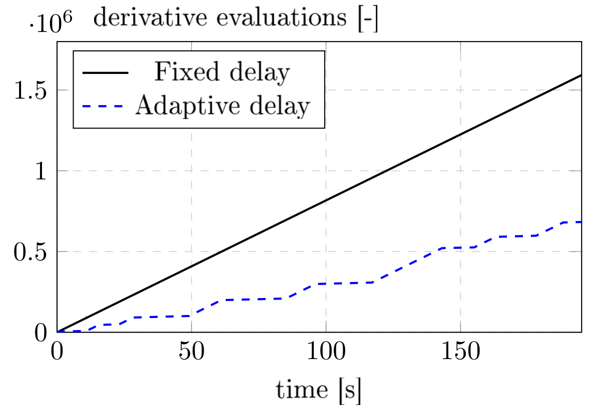

Fig. 15 Number of derivative evaluations for the car and motor sub-models with $\kappa_1 = \kappa_2 = 20$

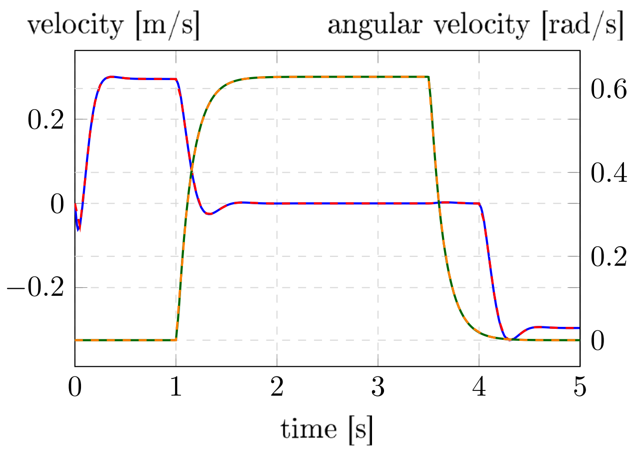

Fig. 16 Simulation results from the robot model with and without delay-size control

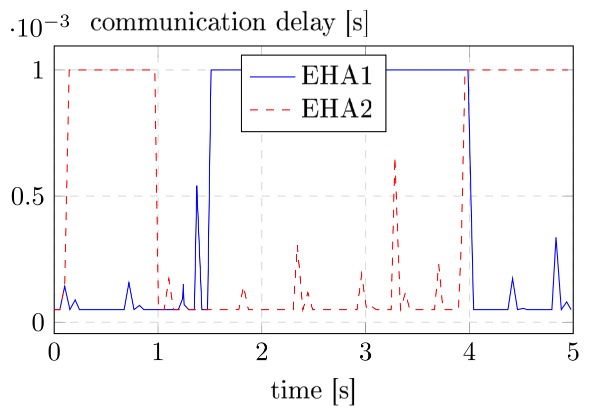

Fig. 17 Communication delays for the two EHA sub-models with $\kappa_1 = \kappa_2 = 20$

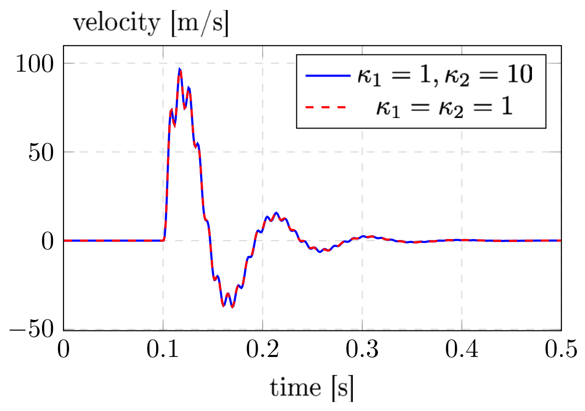

Fig. 19 Velocity in the spring-mass model with and without asymmetric delay-size control

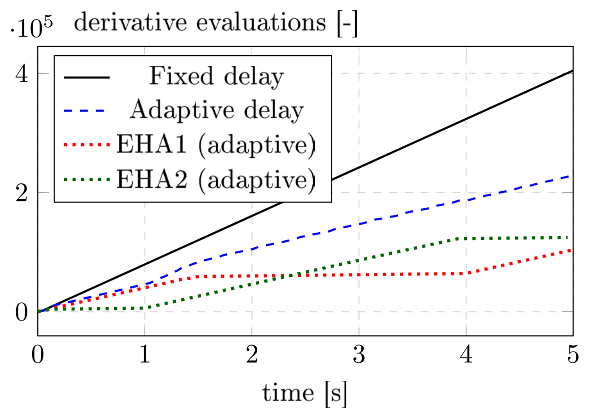

Fig. 18 Number of derivative evaluations for the two EHA sub-models with $\kappa_1 = \kappa_2 = 20$

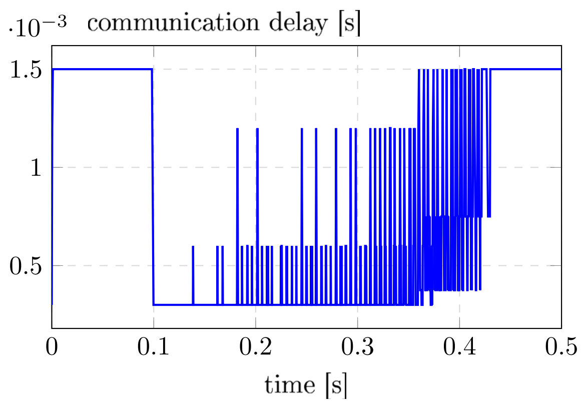

Fig. 20 Communication delay for sub-model 2 with $\kappa_1 = 1$ and $\kappa_2 = 10$

Figure 18 shows the number of derivative evaluation for the two EHA sub-models of the robot model. The number of evaluations in the mechanical 3D model is not of interest, because it does not support adaptive delay-size control. The black reference line indicates the number of derivative evaluations without delay-size control ($\kappa_1 = \kappa_2 = 1$). The two solvers need approximately 44% fewer evaluations with delay-size control.

6.3 Asymmetric delays

The simulation results from running the spring-mass model with delay-size control only in sub-model 2 are shown in Fig. 19. No issues regarding accuracy or stability can be observed.

Figure 20 shows the delay-size used by sub-model 2. As before, the delay-size is increased at the beginning and end of the simulation where the model is found to be at steady state. This confirms that sub-models supporting delay-size control can be connected with TLM to those that do not.

7 Conclusions

We show that it is possible in practice to vary the time delays in a TLM connection. To preserve the dynamic properties of the coupling, the characteristic impedance must be adjusted to compensate for the varying time delay. Furthermore, we also demonstrate that asymmetric TLM connections, with different delays in different directions, are feasible. This enables distributed adaptive delay-size control, where each sub-model decides its delay-size by monitoring its state variables to identify steady-state conditions. It also makes it possible to connect sub-models that support delay-size control with those that do not.

Three SSI algorithms were investigated, and all three provided satisfying criteria for delay-size control. In addition, univariate SSI on the flow variable of a TLM connection was analytically and experimentally shown to be sufficient for identifying steady state. This reduces the number of SSI parameters that need to be calibrated by a factor of two regardless of which of the presented traditional SSI methods is employed.

A significant reduction in the number of derivative evaluations was found when comparing simulations with an adaptive step-size to those with a fixed step-size. This is a promising result, because derivative evaluation is usually the most time-consuming task in continuous-time simulation.

The proposed method depends on the tuning of several parameters. This includes parameters for SSI, which are specific for each SSI method, but also the maximum allowable delay increase and the minimum time threshold between two periods of steady state. Unfortunately, some of these parameters are dependent on model properties and thus cannot easily be generalized. This includes, for example, the amplitude tolerance in the RSW method and the standard deviation of the white noise in the F-test and R-test methods. However, because the delay-size control is handled by each sub-model, the parameters can be chosen by the modeler.

A benefit of letting each sub-model independently handle the delay-size control is that the solution becomes independent of the master simulation tool. Any master algorithm that supports interpolation tables for delayed variables will work with the proposed method. This also includes tools supporting the Functional Mock-up Interface (FMI). FMI version 3.0 supports *intermediate update mode*, where the importing tool and the sub-model can exchange variables at any time between communication points [34]. This enables asynchronous TLM co-simulation [27]. Even though intermediate update is not available in FMI 2.0, the behavior can be mimicked by letting each sub-model handle the interpolation of variables [6]. The methods presented in this paper are compatible with both versions of the standard.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00366-023-01791-1.

Acknowledgements This work was funded by AB SKF and Saab AB.

Funding Open access funding provided by Linköping University.

Data availability The data that supports the findings of this study are available within the article and its supplementary materials.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Benedikt M, Drenth E (2017) Relaxing stiff system integration by smoothing techniques for non-iterative co-simulation. In: IUTAM symposium on solver-coupling and co-simulation, Darmstadt, Germany. https://doi.org/10.1007/978-3-030-14883-6_1
2. Benedikt M, Watzenig D, Hofer A (2013) Modelling and analysis of the non-iterative coupling process for co-simulation. Math Comput Model Dyn Syst 19(5):451–470. https://doi.org/10.1080/13873954.2013.784340
3. Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice Hall, Upper Saddle River
4. Fritzson D, Braun R, Hartford J (2018) Composite modelling in 3-D mechanics utilizing transmission line modelling (TLM) and functional mock-up interface (FMI). Model Identif Control 39(3):179–190. https://doi.org/10.4173/mic.2018.3.4
5. Braun R, Krus P (2014) An explicit method for decoupled distributed solvers in an equation-based modelling language. In: Proceedings of the 6th international workshop on equation-based object-oriented modeling languages and tools, pp 57–64
6. Braun R, Hällqvist R, Fritzson (2017) TLM-based asynchronous co-simulation with the functional mockup interface. In: IUTAM symposium on solver-coupling and co-simulation, Darmstadt, Germany
7. Boden H, Carlsson U, Glav R, Wallin H, Åbom M (1999) Ljud Och Vibrationer. Kungliga Tekniska Högskolan, Institutionen för farkostteknik, Stockholm, Sweden
8. Schierz T, Arnold M, Clauß C (2012) Co-simulation with communication step size control in an FMI compatible master algorithm. In: 9th International Modelica conference, Munich, Germany, pp 205–214
9. Kraft J, Meyer T, Schweizer B (2019) Reduction of the computation time of large multibody systems with co-simulation methods. In: Schweizer B (ed) IUTAM symposium on solver-coupling and co-simulation. Springer, Cham, pp 131–152
10. Eguillon Y, Lacabanne B, Tromeur-Dervout D (2022) F3ornits: a flexible variable step size non-iterative co-simulation method handling subsystems with hybrid advanced capabilities. Eng Comput 38(5):4501–4543. https://doi.org/10.1007/s00366-022-01610-z
11. Meyer T, Kraft J, Schweizer B (2021) Co-simulation: error estimation and macro-step size control. J Comput Nonlinear Dyn **16**(4). https://doi.org/10.1115/1.4048944. 041002
12. Leva A, Bartolini A, Maffezzoni C (1998) A process simulation environment based on visual programming and dynamic decoupling. Simulation 71(3):183–193
13. Papadopoulos AV, Leva A (2014) Automating efficiency-targeted approximations in modelling and simulation tools: dynamic decoupling and mixed-mode integration. Simulation 90(10):1158–1176. https://doi.org/10.1177/0037549714547296
14. Feki ABK-E, Duval L, Faure C, Simon D, Gaid MB (2017) Choptrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. Simulation 93(3):185–200. https://doi.org/10.1177/0037549716684026
15. Busch M (2019) Performance improvement of explicit co-simulation methods through continuous extrapolation. In: Schweizer

B (ed) IUTAM symposium on solver-coupling and co-simulation. Springer, Cham, pp 57–80. https://doi.org/10.1007/978-3-030-14883-6_4
16. Müller W, Breitenecker F (2016) An explicit approach for asynchronous step size control in co-simulation. In: ASIM 2016 23. Symposium Simulationstechnik, pp. 75–80. https://doi.org/10.11128/arep.52
17. Jansson A, Krus P, Palmberg J.O (1992) Variable time step size applied to simulation of fluid power systems using transmission line elements. In: Fifth bath international fluid power workshop, Bath, England
18. Pulko SH, Mallik A, Allen R, Johns PB (1990) Automatic timestepping in TLM routines for the modelling of thermal diffusion processes. Int J Numer Model Electron Netw Devices Fields 3(2):127–136. https://doi.org/10.1002/jnm.1660030208
19. Hui SYR, Fung KK, Zhang MQ, Christopoulos C (1993) Variable time step technique for transmission line modelling. IEE Proc A (Sci Meas Technol) 140:299–3023. https://doi.org/10.1049/ip-a-3.1993.0046
20. Hällqvist R (2019) On standardized model integration: automated validation in aircraft system simulation. Licentiate thesis, Linköping University. https://doi.org/10.3384/lic.diva-162810
21. Krus P (2009) Whole mission simulation for aircraft system design and optimization. In: Proceedings of the CEAS 2009 European air and space conference
22. Hällqvist R, Eek M, Braun R, Krus P (2016) Methods for automating model validation: steady-state identification applied on gripen fighter environmental control system measurements. In: Proceedings of the 30th congress of the international council of the aeronautical sciences. international council of the aeronautical sciences, DCC, Daejon, Korea
23. Cao S, Rhinehart RR (1995) An efficient method for on-line identification of steady state. J Process Control 5(6):363–374. https://doi.org/10.1016/0959-1524(95)00009-F
24. von Neumann J, Kent RH, Bellinson HR, Hart BI (1941) The mean square successive difference. Ann Math Stat 12(2):153–162. https://doi.org/10.1214/aoms/1177731746
25. De Cogan D, O'Connor WJ, Pulko S (2005) Transmission line matrix (TLM) in computational mechanics. CRC Press, Boca Raton, Florida
26. Nakhimovski I (2006) Contributions to the modeling and simulation of mechanical systems with detailed contact analyses. PhD thesis, Linköping University, PELAB—Programming Environment Laboratory, The Institute of Technology
27. Braun R, Fritzson D (2022) Numerically robust co-simulation using transmission line modelling and the functional mockup interface. Simulation
28. Ochel L, Braun R, Thiele B, Asghar A, Buffoni L, Eek M, Fritzson P, Fritzson D, Horkeby S, Hällquist R, Åke Kinnander Palanisamy A, Pop A, Sjölund M (2019) OMSimulator-integrated FMI and TLM-based co-simulation with composite model editing and SSP. In: 13th International Modelica conference
29. Xu H, Wu J, Tseng T.L.B (2018) An efficient method for online identification of steady state for multivariate system. In: International manufacturing science and engineering conference, vol. 4: processes. https://doi.org/10.1115/MSEC2018-6565. V004T03A005
30. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS (2005) SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw (TOMS) 31(3):363–396
31. Modelica and the Modelica Association. https://www.modelica.org/. Accessed 25 Apr 2022
32. Fritzson P (2004) Principles of object oriented modeling and simulation with Modelica 2.1. Wiley-IEEE Press, Hoboken. https://doi.org/10.1109/9780470545669
33. ...Fritzson P, Pop A, Abdelhak K, Ashgar A, Bachmann B, Braun W, Bouskela D, Braun R, Buffoni L, Casella F, Castro R, Franke R, Fritzson D, Gebremedhin M, Heuermann A, Lie B, Mengist A, Mikelsons L, Moudgalya K, Ochel L, Palanisamy A, Ruge V, Schamai W, Sjölund M, Thiele B, Tinnerholm J, Östlund P (2020) The OpenModelica integrated environment for modeling, simulation, and model-based development. Model Identif Control Nor Res Bull 41(4):241–295. https://doi.org/10.4173/mic.2020.4.1
34. Junghanns A, Gomes C, Schulze C, Schuch K, Pierre R, Blaesken M, Zacharias I, Pillekeit A, Wernersson K, Sommer T (2021) The functional mock-up interface 3.0-new features enabling new applications. In: Modelica conferences, pp 17–26