

THE EFFECT AND SELECTION OF SOLUTION SEQUENCE IN CO-SIMULATION

Emin Oguz Inci
Jan Croes
Wim Desmet

Cláudio Gomes
Casper Thule
Kenneth Lausdahl
Peter Gorm Larsen

DMMS Core Lab, Flanders Make
Mechanical Engineering, KU Leuven,
Celestijnenlaan 300, B-3001 Leuven, BELGIUM,
{eminoguz.inci,jan.croes,wim.desmet}
@kuleuven.be

Department of Electrical and Computer Engineering,
Aarhus University, Åbogade 34
8200 Aarhus N, DENMARK,
{claudio.gomes,casper.thule,lausdahl,pgl}
@ece.au.dk

ABSTRACT

In non-iterative serial co-simulation, the solution sequence of the systems is ordered either ad-hoc or based on the solution priorities. The selection of the priorities determines which systems approximate the upcoming inputs from connected systems. This input estimation causes an error on the time integration of the states. This article discusses indicators to estimate the extent of input estimation contribution on the overall state calculation error and addresses to the question if the solution sequence has an influence on the state evolution. Moreover, a temporal analysis of this effect is studied to predict when to flip the sequence at an interface between two systems on a linear numerical example. In conclusion, an adaptive algorithm which switches the co-simulation sequence run-time based on the input estimation accuracy is proposed and validated with a parameter study.

Keywords: co-simulation, solution sequence, input estimation, state-space models.

1 INTRODUCTION

Co-simulation is a technique where the systems are divided into subsystems (Kübler and Schiehlen 2000), based on accuracy, engineering domain, or IP protection concerns (Gomes et al. 2018). Each subsystem is solved by a dedicated solver and the input/outputs at the interfacing connections are coupled in run-time. Although the systems are solved independently between data exchange points, the frequency and the type of coupling has a key role in terms of the accuracy of the solution.

In this paper, we will focus on the serial solutions with Gauss-Seidel coupling algorithms (Bastian et al. 2011). In the serial approach, the subsystems are solved sequentially and the subsystems are coupled after each solution. The subsystem that completes its solution and sends the output data to the connecting subsystems. The input data are updated only at the communication points. If the communication step size is long with respect to the frequency of system dynamics, the accuracy of the solution reduces due to the usage of outdated inputs. The problem can be remedied by increasing the frequency of communication at time periods where input/output signals are volatile. However, increasing the number of communication points increases the time to solution significantly.

Another way to reduce the solution error is to predict the input values and their trajectories until the next communication point. An accurate estimation of the inputs will lead to accurate solutions. When there is no information about the input values in the next communication point, the history of inputs is used to extrapolate the upcoming input values. The subsystems solved later in the solution sequence receives the input of the next communication points and able to interpolate the inputs for estimation.

Researchers have shown empirically in (Gomes et al. 2019) that co-simulation results can be highly sensitive to which subsystem is solved first. Earlier, the works study the stability of different co-simulation algorithms (Arnold 2010, Schweizer, Li, and Lu 2015, Busch 2016, Schweizer, Lu, and Li 2016, Andersson 2016, Gomes et al. 2018). In Section 2, we present an empirical evidence of this effect.

In this paper, we show the difficulties of predicting the best solution sequence and propose a new orchestration algorithm that adaptively adjusts the solution sequence to minimize the co-simulation error. The algorithm is restricted to co-simulations involving two subsystems, and it uses our extension of the well-known error estimation technique called Milne’s Device, from traditional numerical simulations (Fatunla 2014). We show that it performs well with a simple, yet representative running example.

In Section 2, we show that it is possible to reduce the error by changing the solution sequence, and introduce our research question: what determines the best sequence in a connection between two subsystems? In Section 3, show that the best sequence changes over time. In Section 4, different error estimators are tested and our adaptive algorithm is proposed. Section 5 describes the related work, and Section 6 summarizes and discusses the main contributions and limitations of our work.

2 THE EFFECT OF THE SOLUTION SEQUENCE SELECTION

In this section, we investigate the effect of the solution sequence on co-simulation error. The user must assign a solution sequence at the beginning of the serial co-simulation. A two-system architecture, with two possible solution sequences, will be considered to simplify the problem and focus the attention on a single connection. The system that is solved first extrapolates the inputs, whereas the system that is solved last interpolates the inputs. The system that interpolates the inputs will produce more accurate results than the other owing to better estimation of the inputs, as Figure 1a illustrates. If the solution sequence is reversed, this trend will also be reversed. The above motivates an important question: are the two algorithms equivalent in terms of introduced co-simulation error? If not, which one is better?

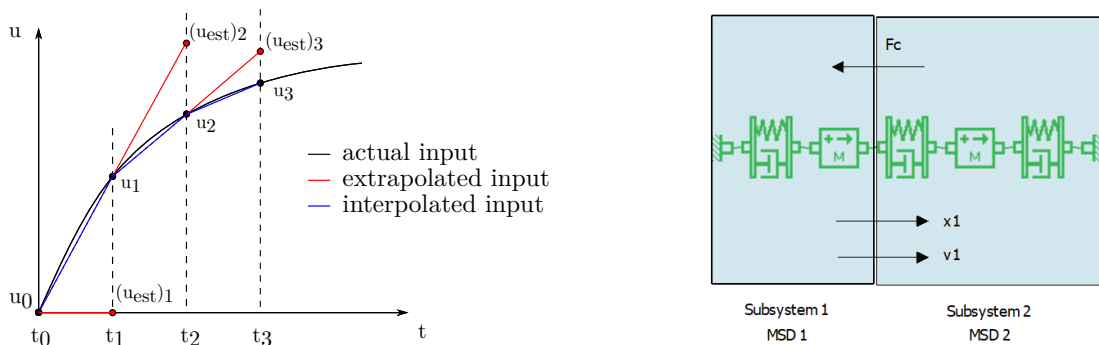


Figure 1: (a) Linear extrapolation vs interpolation, (b) Linear example: two-mass-spring-damper system.

Consider the system consisting of two linear mass-spring-damper subsystems, illustrated in Figure 1b, to make the explanation concrete. The two mass-spring-dampers (MSD) are connected to rigid walls and coupled with a spring-damper. Subsystem 1, MSD1, acts as an inert system by inputting the coupling force from Subsystem 2, MSD2, and outputting displacement and velocity to Subsystem 2. The parameters of the

MSD1 are $m_1 = 1kg$, $c_1 = 1N/m$, $d_1 = 1Ns/m$ denoting the mass, the spring and the damper of subsystem 1, respectively. The parameters of the MSD2 are $m_2 = 1kg$, $c_2 = 1N/m$, $d_2 = 1Ns/m$, $c_c = 1N/m$, $d_c = 1Ns/m$ denoting the mass, the spring and the damper of subsystem 2, and the spring and the damper connecting the two masses, respectively. The given parameter values will be denoted as the parameter set A.

As the springs and the dampers of subsystems are linear, the dynamics of the systems are formulated with a linear state-space representation. This means each sub-system can be formulated as

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad \mathbf{y} = \mathbf{Cx} + \mathbf{Du}, \quad (1)$$

where \mathbf{x}, \mathbf{u} and \mathbf{y} denotes the state, input and output vectors, respectively.

The system can be co-simulated by the two possible solution sequence algorithms: (i) MSD1 \rightarrow MSD2 (MSD1 solved before MSD2), and (ii) MSD2 \rightarrow MSD1, shown in Figure 2. By strongly coupling inputs and outputs of the systems and substituting them into subsystem state-space equations in Equation (1), we can obtain the analytical solution of the system and use it as the reference solution for error calculation. The states of MSD1 and MSD2 are the displacements x_1, x_2 and the velocity v_1, v_2 of the mass on the left and right, respectively. For this given example, the mass on the left is perturbed by $1m$ initially, so the initial conditions of the given example are $x_1 = 1m$, $v_1 = 0m/s$, $x_2 = 0m$, $v_2 = 0m/s$.

Algorithm 1 MSD1 \rightarrow MSD2	Algorithm 2 MSD2 \rightarrow MSD1	
1: doStep(S_1, H)	1: doStep(S_2, H)	At time t:
2: $x_1' \leftarrow \text{getOut}(S_1, x_1)$	2: $F_k' \leftarrow \text{getOut}(S_2, F_k)$	doStep(S_1, H): Calculates the states of S_1 up to time $t + H$
3: $v_1' \leftarrow \text{getOut}(S_1, v_1)$	3: setIn(S_1, F_k, F_k')	getOut(S_1, y): Gets the latest y value in S_1 as output
4: setIn(S_2, x_1, x_1')	4: doStep(S_1, H)	setIn(S_1, u, y): Sets the latest y value to u value in S_1
5: setIn(S_2, v_1, v_1')	5: $x_1' \leftarrow \text{getOut}(S_1, x_1)$	
6: doStep(S_2, H)	6: $v_1' \leftarrow \text{getOut}(S_1, v_1)$	
7: $F_k' \leftarrow \text{getOut}(S_2, F_k)$	7: setIn(S_2, x_1, x_1')	
8: setIn(S_1, F_k, F_k')	8: setIn(S_2, v_1, v_1')	

Figure 2: Possible algorithms.

Figure 3a shows the absolute error in v_1 for the two algorithms shown in Figure 2 with respect to the analytical solution, $\varepsilon_x = |x - x_{ref}|$, and clearly indicates that one algorithm is superior to the other for the given example. Figure 3a motivates the need for choosing the right algorithm, but, as one can see from the results at around time 1.8, there are brief moments where the title of best algorithm changes. In fact, by changing the parameters of the system ($m_1 = 100kg$, $c_c = 100Ns/m$), we obtain results that emphasize the importance of adaptively choosing the right algorithm, see Figure 3b. Let the modified set of parameter values be the parameter set B. The best algorithm with minimum co-simulation error fluctuates at the beginning, but decisively becomes MSD1 \rightarrow MSD2 sequence for the parameter set B after 2.3 seconds.

Not only are the system parameters influential on the results, but also the determination of the initial conditions. If we set the initial conditions differently, such as perturbing the initial location of the m_2 instead of m_1 , $[x_1, x_2] = [0, 1]m$, the state error changes significantly, see Figure 4. In summary, the system parameters and the initial conditions are both highly effective on the results and the selection of the best co-simulation sequence because they change system dynamics.

3 ERROR ESTIMATION

The state errors in the previous section show the need for adaptive master algorithms based on comparing the results of different co-simulation sequences with the analytical solution of the system. In this section,

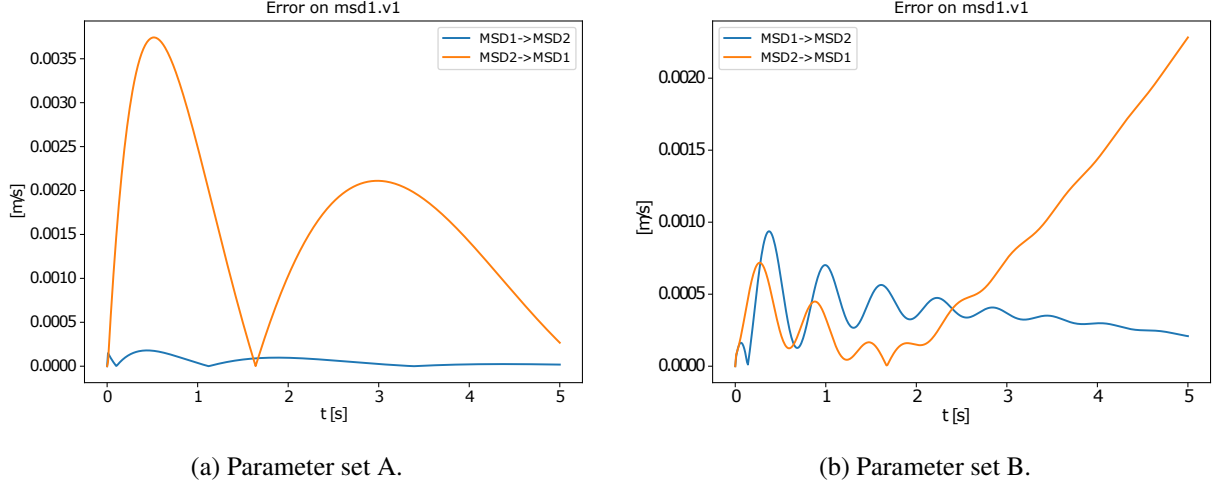


Figure 3: Error in state variable v_1 with respect to different system parameters.

we find an error estimator in order to evaluate how the states are affected by the sequence selection. The input error with respect to the analytical solution is calculated as $\varepsilon_u = |u - u_{ref}|$, where u and u_{ref} are the input value at a certain time in co-simulation solution and analytical solution, respectively.

In order to examine the effect of the inputs to the subsystems, we need to make the input values of the two subsystems comparable. The normalization is an alternative that makes the values dimensionless, and thus comparable, however, the effect of the rate of change in inputs to the evolution of the states cannot be measured by normalization. For this reason, we convert the inputs to a unidimensional external force value for each system, f_{ext} in Equation (2) to compare the input values at the each side of the connection and measure their effect on the states.

$$m_i \ddot{x}_i + d_i \dot{x}_i + c_i x_i = (f_{ext})_i \quad \text{where } i = [1, 2]. \quad (2)$$

The input value for the MSD1 subsystem is the force value directly outputted by the MSD2 subsystem, $F_{direct} = F_c$, and the input value for the MSD2 subsystem is calculated indirectly from the outputted states of MSD1 multiplied with the coupling parameters, $F_{indirect} = d_c v_1 + c_c x_1$. When we calculate the input error for both co-simulation sequences with system parameter set B in Figure 5, a good correlation with state errors is observed in Figure 3b. However, obtaining an analytical solution is either impossible or infeasible in most practical applications. Therefore, we must devise another way to estimate the error.

In the previous section, we indicated that the major difference between the two subsystems is how the inputs are estimated within a communication time step. As the input error with respect to the analytical solution has a good correlation with state errors, we can estimate the co-simulation error with the accuracy of the input estimation for each co-simulation algorithm. As we use linear extrapolation/interpolation through the communication step, the input estimation error is calculated by

$$\varepsilon_n = u_n - \bar{u}_n(u_{n-1}, u_{n-2}), \quad (3)$$

where u_n and \bar{u}_n the actual input and the estimated input at n^{th} communication time step. The linear input estimation, \bar{u}_n , is done in the previous communication point, t_{n-1} by taking u_{n-1} and u_{n-2} into consideration for extrapolation. For interpolation, there is no estimation error because the estimated input is equal to the actual input, $\bar{u}_n = u_n$. Only comparison to be made is between the extrapolation errors of the two solution sequence algorithms MSD1 \rightarrow MSD2 and MSD2 \rightarrow MSD1, see Figure 6a.

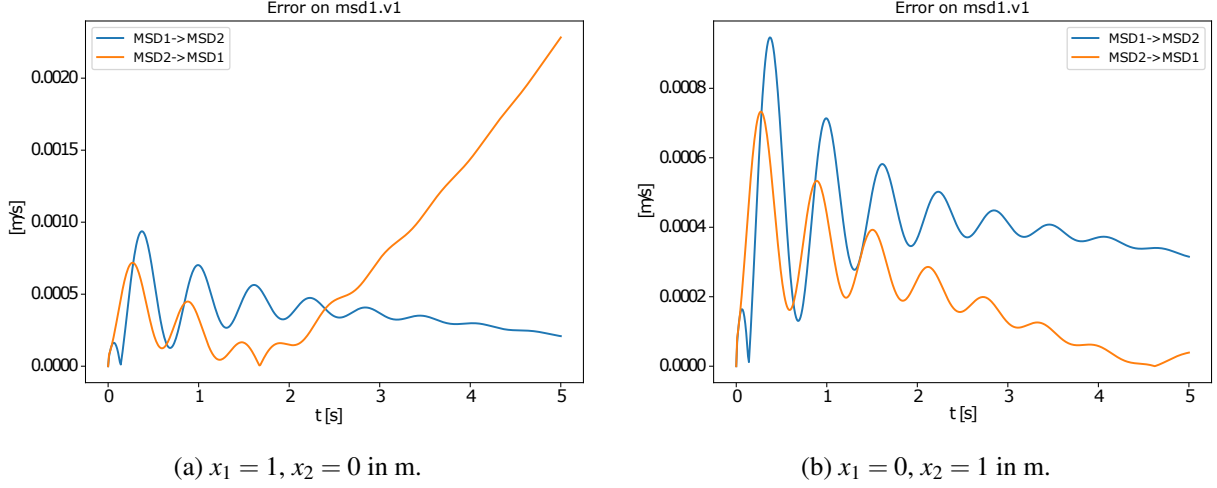


Figure 4: Error in state variable v_1 in parameter set B, with respect to different initial conditions.

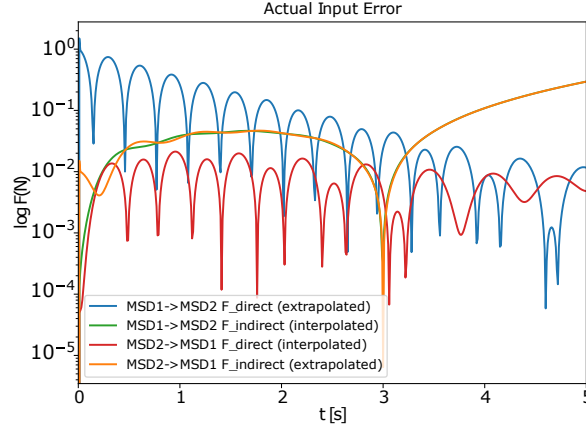


Figure 5: Analytical error of the inputs converted to force.

The above-mentioned input estimation error formula might be a good local estimator, however, it does not estimate how these local errors accumulate and affect state calculation in (1). Therefore, we need a better global estimation technique for co-simulation error. To obtain the global input estimation error, we use a similar technique to global truncation error in (Süli and Mayers 2003) as follows:

$$e_0 = \varepsilon_0, \quad e_1 = \varepsilon_1 H_0 + e_0 J(H_0), \quad \dots \quad e_n = \varepsilon_n H_{n-1} + e_{n-1} J(H_{n-1}), \quad (4)$$

where e_n denotes the global input estimation error at t_n . The parameters H_n and ε_n denotes the communication step size between $[t_n, t_{n+1}]$ and local input estimation error at t_n calculated in Equation (3), respectively. J_n represents the Jacobian at t_n which is the partial derivative of the solution x_{n+1} to Equation (1) over the interval $t_n \rightarrow t_{n+1}$, with respect to input vector, $J(H_n) = \frac{\partial x_{n+1}}{\partial u_n} = BH_n$.

The global input estimation error is shown for the linear example for the parameter set B in Figure 6b. As can be seen, the global error is a better estimate of the error in Figure 3b than the local error. It also provides a better match with the analytical input errors, see Figure 5, when estimating the extrapolation errors. As the input estimation error is fluctuating, a linear Savitzky–Golay filter with a window size of 50 data points is used to smoothen the global input estimation error plot (Savitzky and Golay 1964). The filtered global

input estimation error of MSD2 \rightarrow MSD1 begins to exceed MSD1 \rightarrow MSD2 sequence at $t = 3.4s$ which is fairly close prediction to co-simulation error with $t = 2.3s$.

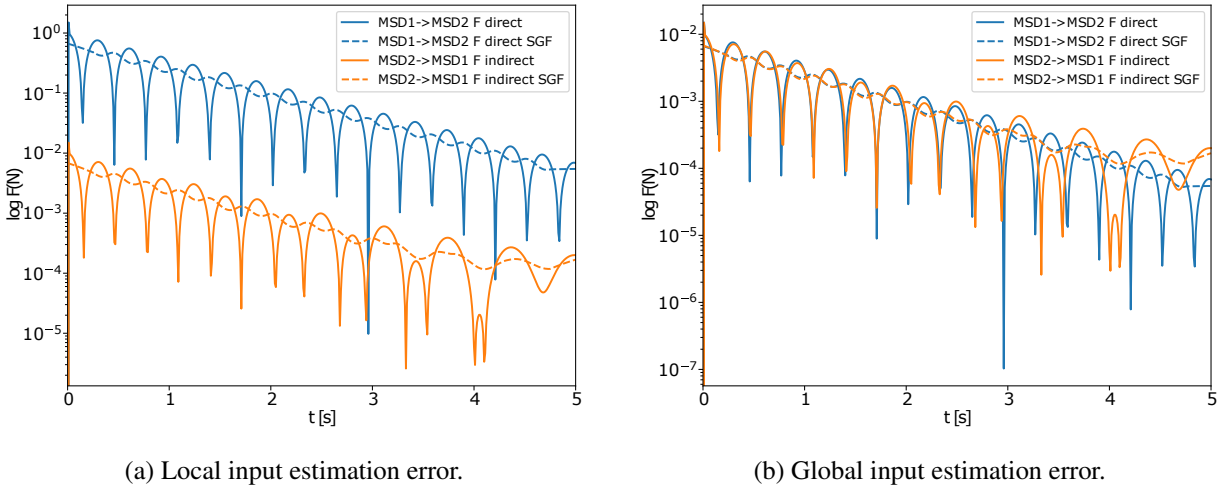


Figure 6: Input estimation error for parameter set B.

4 OPPORTUNITY COST FOR DETERMINATION OF THE SOLUTION SEQUENCE

In the previous section, we presented an error estimator, the global input estimation error, to predict when to switch the solution sequence to reduce the state error. However, the simulation has to be run for each possible co-simulation sequence to be able to predict the sequence and when to switch. We need a methodology that starts from an initial sequence and decides to switch or stay at a co-simulation sequence by estimating the change in accuracy for each scenario. In this section, we will introduce a method that calculates the opportunity cost of switching the sequence and decides the co-simulation sequence at run-time.

The input estimation error is calculated only at the first solved subsystem that extrapolates the inputs for estimation. For the subsystem solved last, which interpolates the inputs within communication steps, a hypothetical error is calculated as if the subsystem were extrapolating the inputs, see Figure 7. With the help of this error, we can evaluate and compare the opportunity cost of the scenarios, to switch or to stay.

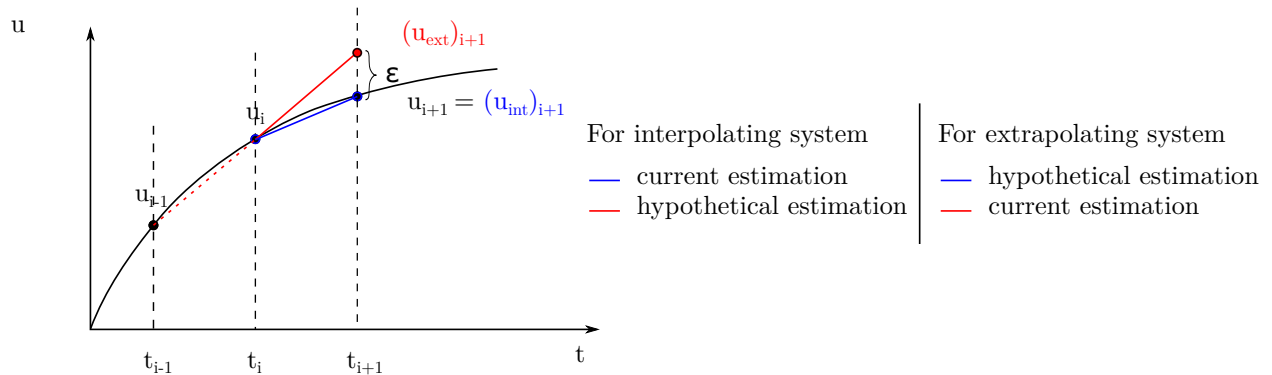


Figure 7: Hypothetical input estimation error when switching the sequence.

Suppose that we select MSD1 \rightarrow MSD2 solution sequence in linear example in ???. In this case, MSD1 subsystem extrapolates the force input, F_c , from MSD2 subsystem and MSD2 subsystem interpolates the

displacement vector, $[x_1, v_1]$, from MSD1 subsystem. For this solution sequence, the actual input estimation error comes from the extrapolation error inflicted at the coupling force, F_c . The selected hypothetical error occurs at the displacement vector assuming that we switch the solution sequence. To make the actual and the hypothetical error comparable, we convert the motion values to forces by multiplying with the coupling parameters as in Equation (2).

The input estimation errors are calculated based on the opportunity cost approach for each co-simulation sequence, see Figure 8. The method is validated as the hypothetical error matches with the actual of the other co-simulation sequence for both algorithms. The method gives fluctuating advice between the two algorithms at the beginning, but it leans towards the MSD1 \rightarrow MSD2 sequence after 4 s.

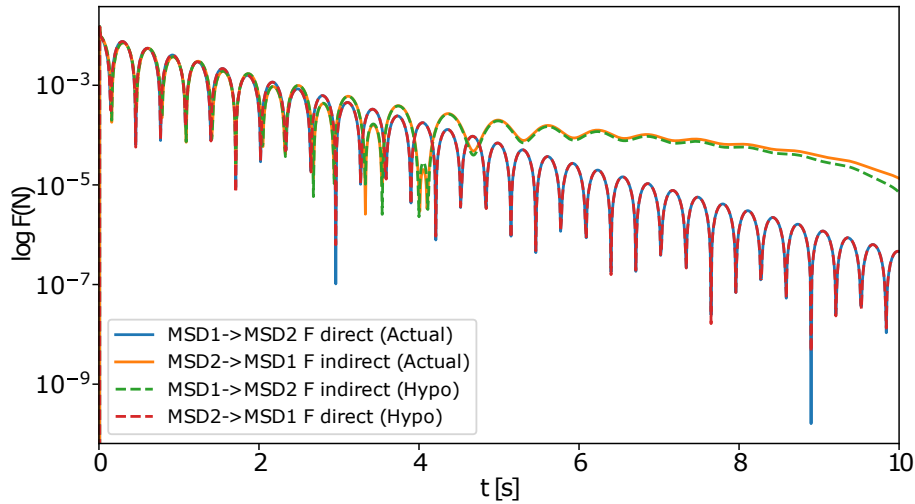


Figure 8: Opportunity cost calculation for each co-simulation sequence.

An adaptive algorithm that switches the co-simulation sequence is developed based on the accuracy of the estimators shown in Section 3. The algorithm compares the actual and the hypothetical error at each communication point and switches to the alternative sequence if the hypothetical error is lower than the actual error. The switching operation is done by determining the priority of the subsystems at each end of the connection and passing the solution sequence to the master orchestrator, see Figure 9. When we run the linear example with the adaptive algorithm for the parameter set B, the results show that the adaptive algorithm switches to the solution sequence with lower state error and avoids instability, see Figure 10.

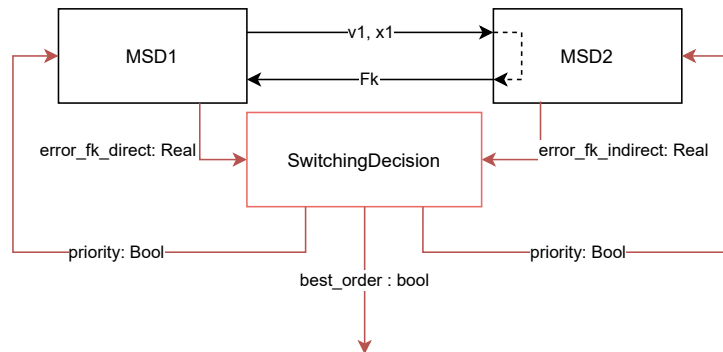
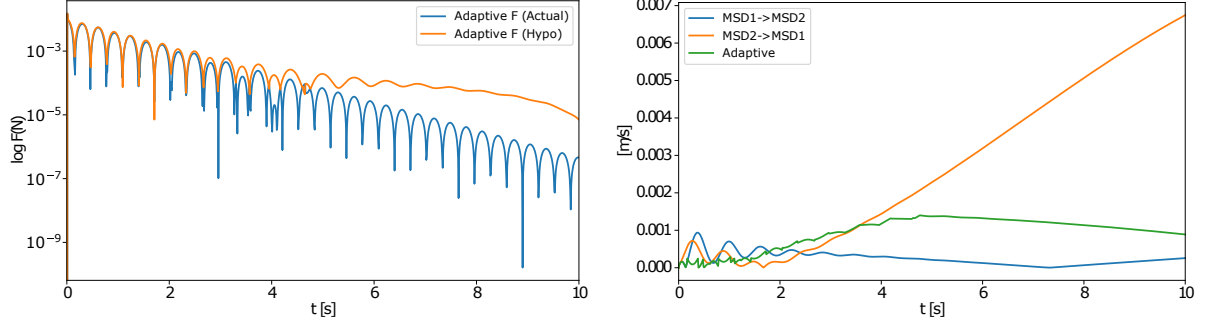
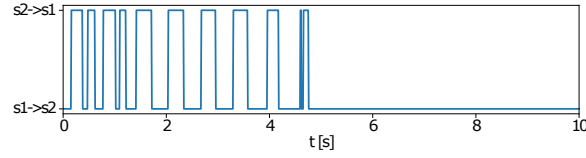


Figure 9: Adaptive algorithm that switches the co-simulation sequence at a connection.



(a) Actual and hypothetical extrapolation errors for the adaptive algorithm. (b) Error in v_1 for adaptive and static co-simulation sequences.



(c) Sequence changes of the adaptive co-simulation.

Figure 10: Adaptive co-simulation sequence algorithm for parameter set B.

Considering the effect of the system parameters and initial conditions on the best co-simulation sequence, we did an extensive parameter study to validate the adaptive algorithm. In order to cover a wide spectrum, each parameter was randomly sampled three times, between $[0.1, 1.0]$ (for masses) or $[0, 1.0]$ (for other system parameters), totalling $3^8 = 6,561$ experiments. Each experiment comprises three co-simulations: two with static alternative solution sequences, and one with the adaptive algorithm. The analytical error is computed once as a reference for the scoring.

The accuracy of the best co-simulation sequence selected by the adaptive algorithm at each time step is scored by the following formula

$$\begin{aligned}
 (\mathcal{E}_{sol})_n &= |(v_{1sol})_n - (v_{1ref})_n| \\
 (\mathcal{E}_{min})_n &= \min((\mathcal{E}_{sol})_n), \quad \text{where } sol = [\text{MSD1} \rightarrow \text{MSD2}, \text{MSD2} \rightarrow \text{MSD1}, \text{adap}] \\
 (\mathcal{E}_{max})_n &= \max((\mathcal{E}_{sol})_n), \quad \text{where } ref = [\text{analytical}] \\
 \text{score} &= \text{mean} \left(\frac{(\mathcal{E}_{max})_n - (\mathcal{E}_{adap})_n}{(\mathcal{E}_{max})_n - (\mathcal{E}_{min})_n} \right).
 \end{aligned} \tag{5}$$

The scoring is maximum (=1) when the adaptive algorithm always selects the best co-simulation sequence.

We kept the score of the 6561 simulations with the adaptive algorithm using the local and global input estimation error as the indicator of the state errors. The mean and median score using the global input estimation error are 0.985 and 0.987, respectively, with a standard deviation of $9.4e-3$, whereas the mean and median score using the local input estimation error are 0.154 and 0.136, respectively, with a standard deviation of $9.9e-2$. The parameter study results show that although the adaptive algorithm using local input estimation error performs poorly, the adaptive algorithm using global input estimation error successfully selects the best solution sequence at any time during the simulation.

5 RELATED WORK

Many works exist that share our goal: reducing the co-simulation error by changing the co-simulation sequence or communication step size. We can classify these works into two main sets: the algorithms that adapt depending on the co-simulation error measured during the co-simulation (adaptive), and those that optimally configure the co-simulation settings before running it (non-adaptive).

The majority of adaptive co-simulation methods include master algorithms that adjust the communication step size based on some error metric, in contrast to our algorithm which adjusts the co-simulation sequence. They can be grouped according to the approach that they follow to estimate the error:

Richardson extrapolation: The essential idea is to compare the results of two co-simulations: one that runs twice the frequency of the other (Arnold et al. 2014, Arnold et al. 2014, Galtier et al. 2015).

Multi-Order Input Extrapolation: The outputs of two different order input approximation methods are compared (Busch and Schweizer 2011, Inci et al. 2019).

Parallel Embedded Method: Runs a traditional adaptive step size numerical method in parallel with the co-simulation (Hoepfer 2011). The purpose is to piggyback on the auxiliary method for the decisions on the step size.

Conservation Laws: The local error is estimated based on the deviation from a conservation law in physics at power bonds. The magnitude of the energy residual at the start and at the end of a co-simulation step serves as an estimate of the local error (Sadjina and Pedersen 2016, Benedikt et al. 2013). An advantage shared with our proposed technique is that it does not require a rollback.

In the category of non-adaptive master algorithms, notable works are presented in (Benedikt and Holzinger 2016, Holzinger and Benedikt 2019b, Holzinger and Benedikt 2019a, Oakes et al. 2021, Gomes et al. 2019, Hansen et al. 2021). Here, the main theme is to formulate a constrained optimization problem of which the cost function is an approximation of the error made by a specific choice of co-simulation algorithm. While these works differ from our work, it is worth noting that they can be used with our proposed method to combine some offline reasoning with online adaptation.

6 CONCLUSION AND DISCUSSION

We showed that the solution sequence of a serial co-simulation has a high impact on the accuracy of the simulation and that the best solution sequence is not static, but changes over simulated time. These phenomena are caused by the selection of the input estimation method, which system is interpolating or extrapolating. Therefore, we used input estimation error as an indicator of co-simulation error. In order to take the input estimation error into account in state calculation and accumulate the effect of the input estimation error, we formulate the global input estimation error. Using this error indicator, we developed an adaptive algorithm that switches the solution sequence by comparing the actual error and the hypothetical error. A parameter study is carried out in a large set of simulations to validate the robustness and the accuracy of the proposed algorithm with different system parameter values. The results show that the proposed algorithm has high accuracy switching to the best solution sequence consistently. The reproducibility package for this paper is available online¹.

Our proposed algorithm, implemented in the MaestroV2 framework (Thule et al. 2020)², always takes a predictable amount of wall clock time to advance in simulated time. The disadvantage is that there is a minimum bound on the error because changing the order does not replace the benefits of adjusting the communication step size. Hence, a practical implementation would use to combine both kinds of algorithms.

¹<https://overture.au.dk/digitaltwin/papers/2021.ANNSIM.AdaptiveMasterMSDResults.zip>

²<https://github.com/INTO-CPS-Association/maestro>

Moreover, the error calculation requires the nature of the couplings between subsystems to be known, and that the Jacobian of the subsystems are provided.

We only focused on the solution sequence at a single interface between two subsystems. As future work, we plan on tackling realistic co-simulation scenarios with multi-physics subsystems, and extending the joint spectral radius analysis to study the stability of adaptive MAs, as was done in (Gomes et al. 2018, Gomes et al. 2018).

ACKNOWLEDGMENTS

Internal Funds KU Leuven are gratefully acknowledged for their support. This research was partially supported by Flanders Make, the strategic research centre for the manufacturing industry. We are grateful to the Poul Due Jensen Foundation, which has supported the establishment of a new Centre for Digital Twin Technology at Aarhus University. We are grateful to the reviewers who dedicated their time and effort to review and improve this paper.

REFERENCES

- Andersson, C. 2016. *Methods and Tools for Co-Simulation of Dynamic Systems with the Functional Mock-up Interface*. Ph. D. thesis, Lund University.
- Arnold, M. 2010, May. “Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models”. *Journal of Computational and Nonlinear Dynamics* vol. 5 (3), pp. 9.
- Arnold, M., C. Clauß, and T. Schierz. 2014. “Error Analysis and Error Estimates for Co-Simulation in FMI for Model Exchange and Co-Simulation v2.0”. In *Progress in Differential-Algebraic Equations*, pp. 107–125. Berlin, Heidelberg, Springer Berlin Heidelberg.
- Arnold, M., S. Hante, and M. A. Köbis. 2014, December. “Error Analysis for Co-Simulation with Force-Displacement Coupling”. *PAMM* vol. 14 (1), pp. 43–44.
- Bastian, J., C. Clauß, S. Wolf, and P. Schneider. 2011, June. “Master for Co-Simulation Using FMI”. In *8th International Modelica Conference*, edited by C. Clauß, pp. 115–120. Dresden, Germany, Linköping University Electronic Press, Linköpings universitet.
- Benedikt, M., and F. R. Holzinger. 2016, April. “Automated Configuration for Non-Iterative Co-Simulation”. In *17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, pp. 1–7. Montpellier, IEEE.
- Benedikt, M., D. Watzonig, J. Zehetner, and A. Hofer. 2013, June. “NEPCE-A Nearly Energy Preserving Coupling Element for Weak-Coupled Problems and Co-Simulation”. In *V International Conference on Computational Methods for Coupled Problems in Science and Engineering, Coupled Problems*, edited by S. Idelsohn, M. Papadrakakis, and B. Schrefler, pp. 1–12. Ibiza, Spain.
- Busch, M. 2016, September. “Continuous Approximation Techniques for Co-Simulation Methods: Analysis of Numerical Stability and Local Error”. *Journal of Applied Mathematics and Mechanics* vol. 96 (9), pp. 1061–1081.
- Busch, M., and B. Schweizer. 2011. “An Explicit Approach for Controlling the Macro-Step Size of Co-Simulation Methods”. In *7th European Nonlinear Dynamics*, edited by D. Bernardini, G. Rega, and F. Romeo, pp. 24–29. Rome, Italy, European Mechanics Society.
- Fatunla, S. O. 2014. *Numerical Methods for Initial Value Problems in Ordinary Differential Equations*. Academic Press.

- Galtier, V., S. Vialle, C. Dad, J.-P. Tavella, J.-P. Lam-Yee-Mui, and G. Plessis. 2015. "FMI-Based Distributed Multi-Simulation with DACCOSIM". In *Spring Simulation Multi-Conference*, edited by H. Samuelson, S. Bhooshan, and R. Goldstein, pp. 804–811. Alexandria, Virginia, USA, Society for Computer Simulation International San Diego, CA, USA.
- Gomes, C., R. Jungers, B. Legat, and H. Vangheluwe. 2018. "Minimally Constrained Stable Switched Systems and Application to Co-Simulation". Technical Report arXiv:1809.02648, University of Antwerp, Belgium.
- Gomes, C., B. Legat, R. Jungers, and H. Vangheluwe. 2018. "Minimally Constrained Stable Switched Systems and Application to Co-Simulation". In *IEEE Conference on Decision and Control*, pp. 5676–5681. Miami Beach, FL, USA.
- Gomes, C., B. J. Oakes, M. Moradi, A. T. Gamiz, J. C. Mendo, S. Dutre, J. Denil, and H. Vangheluwe. 2019. "HintCO - Hint-Based Configuration of Co-Simulations". In *International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, edited by M. Obaidat, T. Ören, and H. Szczerbicka, pp. 57–68. Prague, Czech Republic.
- Gomes, C., C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe. 2018. "Co-Simulation: A Survey". *ACM Computing Surveys* vol. 51 (3), pp. 49:1–49:33.
- Gomes, C., C. Thule, K. Lausdahl, P. G. Larsen, and H. Vangheluwe. 2018. "Stabilization Technique in INTO-CPS". In *2nd Workshop on Formal Co-Simulation of Cyber-Physical Systems*, Volume 11176. Toulouse, France, Springer, Cham.
- Hansen, S. T., C. Thule, and C. Gomes. 2021. "An FMI-Based Initialization Plugin for INTO-CPS Maestro 2". In *Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops*, edited by L. Cleophas and M. Massink, pp. 295–310. Virtual event, Springer International Publishing.
- Hoepfer, M. 2011. *Towards a Comprehensive Framework for Co-Simulation of Dynamic Models With an Emphasis on Time Stepping*. Ph. D. thesis, Georgia Institute of Technology, USA.
- Holzinger, F., and M. Benedikt. 2019a. "Optimal Trigger Sequence for Non-Iterative Co-Simulation:". In *Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, edited by M. Obaidat, T. Ören, and H. Szczerbicka, pp. 80–87. Prague, Czech Republic, SCITEPRESS - Science and Technology Publications.
- Holzinger, F. R., and M. Benedikt. 2019b. "Hierarchical Coupling Approach Utilizing Multi-Objective Optimization for Non-Iterative Co-Simulation". In *13th International Modelica Conference 2019*, edited by A. Haumer, pp. 6. Regensburg, Germany, Linköping University Electronic Press.
- Inci, E. O., J. Croes, J. Bosmans, M. Vermaut, and W. Desmet. 2019. "Adaptive Co-simulation Framework Exploiting System Physics of Mechatronic Systems". In *International Conference on Adaptive Modeling and Simulation*. Alicante, Spain.
- Kübler, R., and W. Schiehlen. 2000. "Two Methods of Simulator Coupling". *Mathematical and Computer Modelling of Dynamical Systems* vol. 6 (2), pp. 93–113.
- Oakes, B. J., C. Gomes, F. R. Holzinger, M. Benedikt, J. Denil, and H. Vangheluwe. 2021. "Hint-Based Configuration of Co-Simulations with Algebraic Loops". In *Simulation and Modeling Methodologies, Technologies and Applications*, Volume 1260, pp. 1–28. Cham, Springer International Publishing.
- Sadjina, S., and E. Pedersen. 2016, June. "Energy Conservation and Coupling Error Reduction in Non-Iterative Co-Simulations". Technical report, SINTEF Alesund.
- Savitzky, A., and M. J. Golay. 1964. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures.". *Analytical chemistry* vol. 36 (8), pp. 1627–1639.

- Schweizer, B., P. Li, and D. Lu. 2015, September. “Explicit and Implicit Cosimulation Methods: Stability and Convergence Analysis for Different Solver Coupling Approaches”. *Journal of Computational and Nonlinear Dynamics* vol. 10 (5), pp. 051007.
- Schweizer, B., D. Lu, and P. Li. 2016, January. “Co-Simulation Method for Solver Coupling with Algebraic Constraints Incorporating Relaxation Techniques”. *Multibody System Dynamics* vol. 36 (1), pp. 1–36.
- Süli, E., and D. F. Mayers. 2003. *An Introduction to Numerical Analysis*, pp. 317–343. Cambridge University Press.
- Thule, C., M. Palmieri, C. Gomes, K. Lausdahl, H. D. Macedo, N. Battle, and P. G. Larsen. 2020. “Towards Reuse of Synchronization Algorithms in Co-Simulation Frameworks”. In *Software Engineering and Formal Methods*, Volume 12226 of *Lecture Notes in Computer Science*, pp. 50–66. Oslo, Norway, Springer International Publishing.

AUTHOR BIOGRAPHIES

EMIN OGUZ INCI is a research associate in Mechanical Engineering Department at KU Leuven. He holds an M.Sc. from Computational Mechanics at TU Munich and an B.Sc. from Mechanical Engineering at METU. His research interests include mechatronic design, simulation of multi-physical systems and design space exploration. eminoguz.inci@kuleuven.be.

CLÁUDIO GOMES is a postdoc researcher at Aarhus University (Denmark). He received his PhD at the University of Antwerp, for his work on the foundations of co-simulation. His web address is <https://pure.au.dk/portal/da/claudio.gomes@ece.au.dk>.

JAN CROES is a post-doctoral researcher at KU Leuven in the department of Mechanical Engineering. He predominantly focusses on application oriented research in the field of system level modelling of mechatronic drivelines in combination with model based sensing techniques jan.croes@kuleuven.be.

WIM DESMET is a full professor at the Department of Mechanical Engineering and serve as the Managing Director of KU Leuven. His research specialises in mechatronic system dynamics and vibroacoustics. Professor Desmet has been chair of the IOF Council at KU Leuven since 2015. He also leads a research group at Flanders Make, the strategic research centre for the manufacturing industry. wim.desmet@kuleuven.be

CASPER THULE is a PostDoc at Aarhus University and member of the AU DIGIT Centre of Digitalization, Big Data and Data Analytics. His area of research is hybrid co-simulation and cyber-physical systems. casper.thule@ece.au.dk

KENNETH LAUSDAHL is a researcher at Aarhus University. He holds a PhD from AU with focus on tool automation for formal methods and co-simulation. He has been involved in research projects in the CPS and co-simulation domain during the last decade. lausdahl@ece.au.dk

PETER GORM LARSEN is a professor at Aarhus University. He has been involved with >40 research projects and he currently leads the AU DIGIT Centre, the AU Centre for Digital Twins, as well as the research group for CPSs. He has published more than 200 articles and three books on modelling of CPSs and DTs. pgl@ece.au.dk