Co-simulation and Dynamic Model Exchange with Consideration for Wind Projects

Cvetkovic, Milos; Krishnappa, Harish; López Torres, Claudio; Bhandia, Rishabh; Rueda, José L.; Palensky, Peter

Publication date
2017

Document Version
Peer reviewed version

Published in
Proceedings - 16th Wind Integration Workshop

Citation (APA)
Cvetkovi, M., Krishnappa, H., López, C. D., Bhandia, R., Rueda Torres, J., & Palensky, P. (2017). Co-simulation and Dynamic Model Exchange with Consideration for Wind Projects. In U. Betancourt, & T. Ackermann (Eds.), Proceedings - 16th Wind Integration Workshop: International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants Darmstadt: Energynautics GmbH.

Co-simulation and Dynamic Model Exchange with Consideration for Wind Projects

Miloš Cvetković, Harish Krishnappa, Claudio David López, Rishabh Bhandia, José Rueda Torres, Peter Palensky
Faculty of Electrical Engineering Mathematics and Computer Science
Delft University of Technology
Delft, The Netherlands
Email: m.cvetkovic@tudelft.nl

***Abstract*—This paper discusses and compares two approaches to address technical challenges in performing collaborative studies of power system dynamics. On one side, we consider the *model migration* approach which is an essential piece of dynamic model exchange. On the other side, we look at the *co-simulation* approach which is used to couple simulation tools together. The approaches are compared in terms of accuracy and the most probable reasons for discrepancy between dynamic responses are discussed. We complement the discussion and conclusions with the simulation results on a simple test system.**

I. Introduction

In recent years, ENTSO-E has shown high dedication to standardizing model exchange between TSOs. The Common Grid Model Exchange Standard (CGMES) has emerged as an embodiment of this initiative and the Common Information Model (CIM) as an underlying standard for its implementation. Currently, the CGMES identifies model exchange for dynamic studies as a key area for improvement and further growth. A particular emphasis is on the extensions for handling user-defined models, wind power plants/parks, HVDC lines and FACTS devices.

Besides the foreseen standard improvements, the practical question of the most appropriate vehicle for dynamic model exchange has not yet been decided upon. The main challenge to overcome stems from the fact that the TSOs keep the models of their system encapsulated within commercial simulation tools. Since different TSOs use simulation tools of different vendors, the problem of finding the appropriate means for model exchange becomes the problem of finding an adequate technical solution for sharing the information between simulation tools.

One possible approach to dynamic model exchange is to use a domain independent modeling and simulation language, such as Modelica, as a target platform for model migration. As a matter of fact, the latest version of CGMES v2.5 proposes Modelica for dynamic model exchange of user defined models [1]. Some initial studies on systematic binding of Modelica classes from the iTesla Power System Modelica Library (iPSL) and the Open-Instance Power System Library (OpenIPSL) with the CGMES-CIM have been already performed [2]. This binding should allow for automated model migration while simplifying the process for the user.

The main challenge of this approach is that the commercial tools for simulation of power system dynamics often deploy proprietary modifications of numerical solvers in order to improve the performance of their tool. At times, these proprietary modifications tinker with the model, thus blurring the line between the model and the solver. Hence, there are no guarantees that the domain independent tool will produce exactly the same shape of the time response curve as the commercial simulation tool.

Simulations of wind-enriched power systems entail further challenges for the domain independent tool approach. Since the converter controller models are often kept as black box models, it is impossible to migrate them to a domain independent tool without the involvement of the manufacturer. Another difficulty is that the user defined models which are often found in large expansion projects such as integration of wind farms, are technologically challenging to migrate to a domain independent tool.

The concept of combined simulation (co-simulation) is an alternative to model migration. A co-simulation is created by exchanging data between the commercial simulators at runtime [3]. Instead of migrating the models, the commercial simulators are coupled through available application programming interfaces (APIs) and engaged in a simulation. This approach guarantees that even the vendor-made adjustments to the solver and the model would be included in the resulting case study. The co-simulation, of course, does not come without practical challenges of its own.

In this paper, we seek to compare the pros and cons of two approaches: 1) the model migration to a domain independent modeling and simulation tool (OpenModelica in particular), and 2) the co-simulation between instances of DIgSILENT PowerFactory. PowerFactory is a widely accepted tool of choice in the power industry. Since the current versions of iPSL and OpenIPSL lack PowerFactory-derived models, the models developed and validated in [4] are used.

The comparison is firstly done at a conceptual level discussing advantages and disadvantages of the two approaches, followed by the comparison of the actual simulation results. The example test case, rather simplified for tractability, is taken from the domain of wind integration.

The rest of the paper is organized as follows. In Section II we introduce the intricacies of the process of model migration. In Section III we describe the co-simulation development process. In Section IV we compare both approaches in action using a simple grid model. Section V concludes the paper.

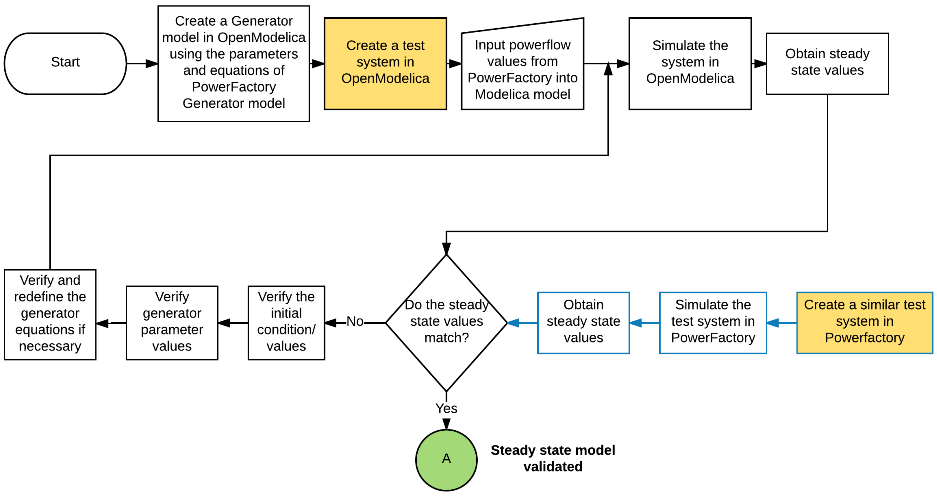

Fig. 1. The first part of model migration: Process of model validation for steady state models.

II. Model Migration

Exchanging power system model information, particularly for simulation of user defined models, is technically challenging and impedes collaboration between different electricity grid stakeholders [5]. The use of various simulation software tools with their corresponding and often incompatible data formats, together with the use of models that are strongly coupled to the software numerical solvers are the main roadblocks for performing collaborative studies of power system dynamics. Although the most common power system simulation tools are computationally efficient and reasonably user-friendly, they also have closed architectures [6], which makes it difficult to thoroughly understand the numerical integration process behind the scenes. The use of open source software for simulation and modeling can help to bridge the differences between commercial simulation tools and enhance coordination between the stakeholders.

In recent years, the non-proprietary modeling language Modelica has been gaining momentum in the power system community. The Modelica language was introduced in 1997 and since then has found its use for the studies of dynamics in various fields, including the automotive industry, thermo-fluid and energy system modeling, and mechatronics. Each of these expert domains has produced libraries of components targeted at the domain specific applications. In the power system community, such effort is spearheaded by the FP7 iTesla project [7].

Modelica is an object oriented, equation-based modeling language that puts emphasis on two features. First, the models of physical devices are specified using differential and algebraic equations. Such natural system description makes it easy for domain experts from many fields to easily code mathematical descriptions of their systems using Modelica. The second important feature is Modelica's focus on separating the model description from the numerical solver. Hence, the domain expert has the full control over the simulation method deployed. Some of the major engineering tools, such as Matlab and Mathematica, provide interfaces for the Modelica language.

Currently, two main power system libraries for Modelica exist, iPSL and OpenIPSL [8]. The libraries contain typical power system components, such as network elements, power generation technologies and aggregated consumption models. The models are developed to replicate the behavior of the models within specific software platforms like PSAT, PSS/E and Eurostag. All of the models are based on RMS values.

In this paper, PowerFactory is used as a commercial reference tool. Since the PowerFactory-based models are currently not available within iPSL or OpenIPSL, the RMS models in Modelica language developed in [4] are used in this paper. These models cover more complex devices (e.g. synchronous generators). Models of the electric buses, transmission lines and loads are directly taken from OpenIPSL. In addition to these models, OpenModelica Integrated Development Environment is used to create a case system in Modelica language.

A. *The Process of Model Migration*

Migrating models from PowerFactory into the Modelica language is a challenging task since both tools are built on different premises. One possible approach is described next. First, one has to identify the component equations that drive the component dynamics by reading various textbooks, user manuals and help documentation. Next, the initialization equations must be identified, typically using empirical studies. The model is then written in the Modelica language.

In order to test and validate a component model, a test system needs to be created both in OpenModelica and in PowerFactory. In this approach to model migration, the steady state models are validated followed by the dynamic model validation. Both processes are respectively illustrated in Fig. 1 and Fig. 2 on an example of the synchronous generator component.

The steady state model validation starts with inserting an initial guess for steady state simulation in OpenModelica (power flow values). Next, the responses from both platforms are compared. If the settling values of all comparable variables match, then the initialization equations are correct and the developer can proceed with the dynamic response

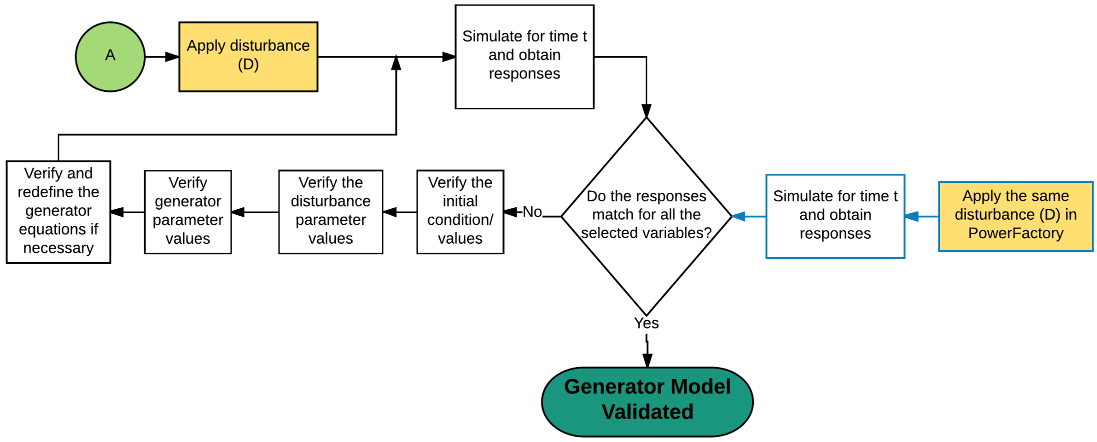

Fig. 2. The second part of model migration: Process of model validation for dynamic models.

validation. Otherwise, the initial condition equations, initial guesses and/or parameters should be revised.

In the dynamic model validation process, a disturbance is added in the Modelica grid model and the model is simulated. The same kind of disturbance is added in PowerFactory and the responses of all variables are monitored. They are subsequently compared and if they do not match satisfactorily, then the model equations and parameter values need to be revised. This trial-and-error process is continued until the responses match. The process remains the same for developing other component models.

B. Some Probable Causes of Differences Between Responses

1) Models: Gathering model information is the key to model migration. Typically, the documentation of the commercial tools is prepared with the intent of supporting the user as a domain expert in utilizing the tool for their needs. Therefore, it could easily happen that quite a few details required for model development/migration are missing from the documentation.

While user-defined models are easy to implement thanks to the full transparency, some of the closed models pose a much greater challenge. The functionality of these models is typically encapsulated as proprietary while the supporting documentation is insufficient to deduce their mathematical description (an example is the PowerFactory OLTC transformer [4]). In the case of such components, one must resort to creating approximate models.

2) Solvers: In addition to matching the models, the solvers used in both the reference and the destination simulation tool must be the same. Since the exact solver used by PowerFactory is not known to the authors of this paper, an OpenModelica solver that provided the best match on a set of case studies was chosen. This solver is a hybrid nonlinear solver Radau 5 [9]. Using an empirical trial-and-error procedure, it was found that this numerical solver is the best match for the PowerFactory solver when the criteria is to match dynamic system responses as best as possible.

3) Initialization: While it is relatively easy to find differential and algebraic state equations for most of the components that constitute a model, identifying the equations that are used for initialization of the model is a level of difficulty higher. Each commercial tool might use a different kind of initialization mechanism depending on the modeling philosophy followed. Therefore, one of the crucial parts of the model migration is to identify the initialization equations and the method. The most natural approach is to resort to intuition and adjust the initial state values until a satisfying match between responses is achieved.

In the particular case of OpenModelica, the so-called guess values are given by the developer and the OpenModelica initialization tool is started to compute the initial point for the simulation. OpenModelica cannot set the guess values automatically since it is not equipped with a power flow solver of its own, Therefore, a power flow solution must be computed by another tool/method and subsequently loaded into OpenModelica to be used as the guess value. The initialization of Modelica-based power system models is recognized as a open problem. In the future, this process will be simplified by the use of tools such as PSM [10]. For the needs of this paper, the power flow computation is carried out in PowerFactory and the resulting power flow values are loaded into OpenModelica as guess values.

III. Co-simulation Approach

Co-simulation, or coupled simulation, is a term used to describe two or more coupled simulators that are simultaneously exchanging data during the simulation execution [3]. The approach is based on a premise that no single system architect exist, i.e. simulation tools and models were initially developed without consideration of the other tools and models. This premise is very much true in the field of power systems where historically each commercial simulation tool was built independently.

The approach has found the most straightforward use in coupling of simulations from different physical domains (e.g. transmission grid simulation and communication network simulation [11]) and for coupling of EMT and RMS simulations [12]. In the recent work [13], the idea of simulation coupling for cooperation between TSOs and DSOs is explored in more detail.

In this paper, a co-simulation using two instances of PowerFactory is created. A part of the power system model is

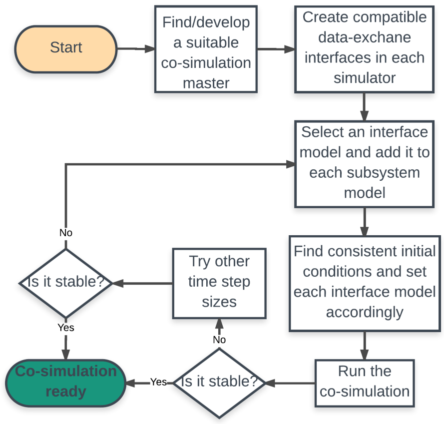

Fig. 3. Process of co-simulation implementation.

kept within one instance while the other part is kept within the other instance. The main reason to couple two PowerFactories is for validation purposes since the comparison with the reference PowerFactory model is only meaningful in this case. Nevertheless, the core-benefits of the co-simulation approach are more prominent in coupling of two different proprietary tools.

To create the co-simulation, PowerFactory is engaged through the use of its application API and the DIgSILENT Simulation Language (DSL) socket blocks [14]. The API is used for configuring and starting the simulation while the DSL-based socket connection is used to exchange state values with every time step. The use of these methods is described in more details in [15].

A. *The Process of Co-simulation Development*

The process of co-simulation requires minimal tampering with the existing models. However, most of the development work is done outside of the power system simulation tool environments as illustrated in Fig 3. The most time consuming part of the process is the development of the master algorithm that is responsible for message exchange and synchronization between the simulators. Next, the so-called interface models have to be developed for each simulation tool. These models constitute equivalent representations of the models within connected simulation tools. In addition, appropriate drivers for accessing the interface variables need to be developed if they are not already available (which is not the case with PowerFactory as previously described) and aligned with the master algorithm. The initial conditions are computed next. This is typically done by iteratively engaging the power flow solvers of all coupled simulators until the convergence of the interface variables is achieved. Finally, the model is co-simulated.

The validation of the responses against a reference simulation would not be possible in the most general case of co-simulation since the reference model would not exist. In this paper, however, we perform the validation against a reference PowerFactory model for the sake of comparison.

It is worth noting that the co-simulation performed in this paper is EMT-based (the simulators run EMT models) while the model migration is performed based on the RMS models. The single reason to opt for an EMT-based co-simulation is the existence of the extensive body of literature on this topic and relatively simpler master and interface implementation. On the other hand, the model migration is simpler to perform with RMS-based models.

B. *Some Probable Causes of Differences Between Responses*

1) Interfaces: The interfaces must remain numerically stable and accurate during steady state and transients. These two properties are difficult to evaluate, especially if the benchmark simulation is unavailable.

The interfaces must be (at least partially) created from the available components within the simulation tools. In the case of electrical systems, this is typically done with the aid of voltage and current sources. However, the use of ideal voltage and current sources can have detrimental consequences on the accuracy and numerical stability of the co-simulation [12]. The solution can be sought by using more complex interface models. For example, a Thevenin equivalent can be used as a replacement for an ideal voltage source. In this paper, the Damping Impedance Method [16] has been used. This method uses two voltage sources and a current source as an interface model.

If the simulators are not designed to be used for co-simulation, which is often the case, creating interfaces could potentially require cumbersome workarounds. These workarounds can have limitations, e.g. in terms of accessible functionalities through API, that affect the quality and efficiency of the co-simulation. As co-simulation standards, like the Functional Mockup Interface (FMI), are adopted by simulation tools, co-simulations will become less challenging.

2) Synchronization: Time synchronization is one of the two essential functionalities of the master algorithm. Each simulator inherently runs at its own rate, produces outputs at each time step, and waits for new inputs before executing the next time step. This procedure results in a palette of time steps for all simulators. The co-simulation master must handle different rates and make sure that each simulator receives the inputs it needs and when it needs them. To achieve this, the co-simulation master would have to accommodate adjustable step sizes and might need to perform interpolation of simulator outputs. More details on the master algorithm used in this paper are found in [13].

3) Initialization: Finding the initial operation point is challenging since it requires synchronization of independently computed power flow calculations. As in [13], an iterative method might be needed to find an operation point that is consistent across co-simulation interfaces. The initialization process increases in complexity if the interface models are complex. In the case of the Damping Impedance Method, several sources need to be set correctly and power flow calculations must converge.

IV. Simulation Results

The test system presented in this paper consists of a simple power system. A round rotor synchronous generator with voltage and power ratings of 20 kV and 100 MVA is connected to a simple distribution grid as shown in Fig. 4.

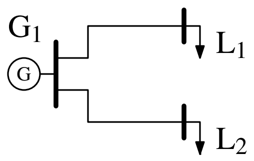

Fig. 4. One-line diagram of the test system.

Two identical 3-phase AC loads are attached via two parallel and identical medium voltage lines.

A small test system is chosen in order to reduce the complexity of the model while still capturing sufficient number of details to demonstrate both approaches. A synchronous generator model is used to roughly represent a wind generator since a Modelica equivalent of the PowerFactory DFIG model was unavailable at the time of this study. A proximity with a low-inertia wind generator is achieved by reducing the inertia constant of the synchronous generator.

PowerFactory 15.2 is used as a reference simulation. In the case of co-simulation, the system is divided so the load L_2 and the line connecting it to the generator are simulated in a separate instance of PowerFactory.

Two test cases have been simulated. In the first test case, a short circuit at the bus with load L_1 of duration 0.2 s occurs at $t = 5$ s. In the second test case, a complete loss of load L_1 occurs at $t = 5$ s and lasts until the end of the simulation.

Although all simulations are performed from $t = 0$ s to $t = 10$ s, the shown plots are zoomed-in so that the most prominent variations in the most representative states can be observed.

Fig. 5 shows the speed of the generator in the case of the loss of load. Two things can be noticed in this plot. First, the equilibrium points for both model migration and co-simulation at time $t = 5$ s are different from the equilibrium points of the benchmark simulations. This is attributed to the error introduced by the co-simulation interface. Second, model migration shows slightly better performance in terms of dynamic tracking of the benchmark simulation.

Fig. 6 shows the electrical torque in the case of the short circuit. This plot nicely illustrates the difference between the EMT and RMS approaches. Both co-simulation and model migration achieve satisfying tracking of their own respective benchmark simulations.

Fig. 7 shows the voltage on the bus of L_2 in the case of the loss of load and Fig. 8 shows the behavior of the voltage on the bus of L_2 in the case of the short circuit. It is visible in both of these plots that the co-simulation achieves slightly better tracking of the benchmark behavior than the model migration following the first few milliseconds after the event. During those first few milliseconds after the event, the co-simulation shows larger differences with respect to benchmark, particularly visible in terms of very short glitches of relatively high magnitude. These differences are attributed to the error introduced by the minimal available step size in the co-simulation approach and to the large value of derivatives in the underlying differential equation models. In the case of model migration, very fast transients are neglected by the RMS model and thus, the better tracking during the first few milliseconds after the fault.

Table I and Table II show the Root Mean Square Error

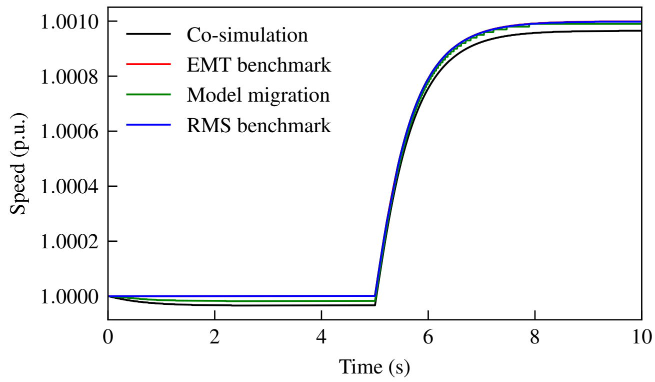

Fig. 5. Generator speed (ω_g) during a total loss of load L_1. Both benchmarks overlap.

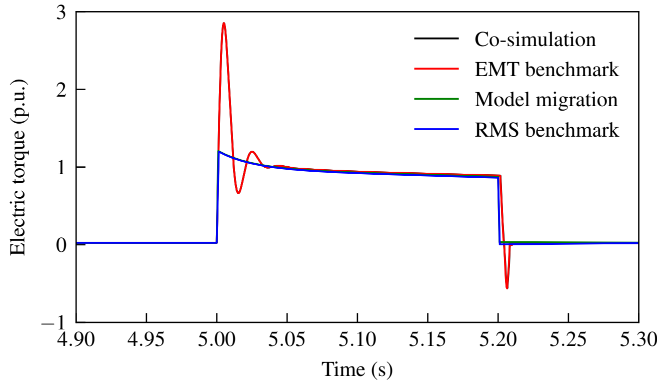

Fig. 6. Generator electric torque (T_e) during a short circuit at load L_1. The results from each method overlap with their respective benchmarks.

(RMSE) of the three variables under observation in the case of short circuit and in the case of loss of load, respectively. The errors are calculated with respect to the results obained from the benchmark simulations. The numbers confirm the conclusions of the visual inspection of the plots. Both approaches match the benchmark simulations well. In general, the RMSE numbers for the co-simulation are slightly higher, which can be traced to the significant contribution of the error in the first few milliseconds after the event.

V. Conclusion

Two technical approaches for collaborative power system dynamic simulation studies have been described in this text, model migration and co-simulation. The critical aspects for

TABLE I
RMSE for a Short Circuit Event

Variable	Model migration	Co-simulation
ω_g	0.000 129 p.u.	0.000 033 p.u.
V_L	0.002 671 p.u.	0.016 173 p.u.
T_e	0.002 243 p.u.	0.001 493 p.u.

TABLE II
RMSE for a Loss of Load Event

Variable	Model migration	Co-simulation
ω_g	13.341×10^{-6} p.u.	32.698×10^{-6} p.u.
V_L	0.000 829 p.u.	0.007 910 p.u.
T_e	0.000 232 p.u.	0.000 462 p.u.

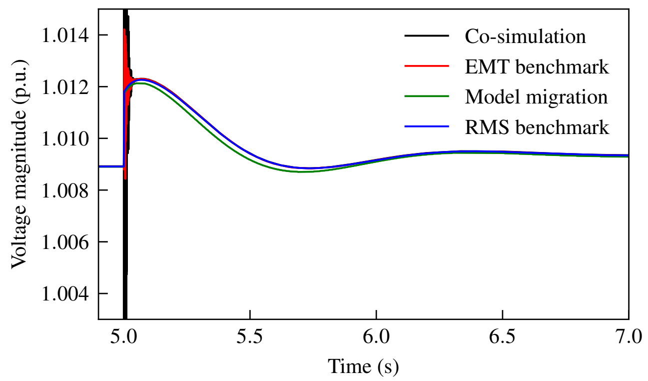

Fig. 7. Voltage magnitude at load L_2 (V_L) during a total loss of load L_1.

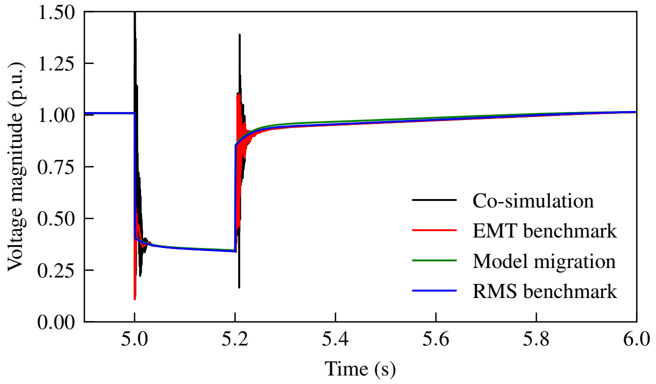

Fig. 8. Voltage magnitude at load L_2 (V_L) during a short circuit at load L_1.

accuracy with respect to the benchmark simulation have been discussed. In the case of model migration, the emphasis is on matching component models, numerical solvers and the initialization procedure. In the case of co-simulation, the focus is placed on utilizing accurate interface models, synchronization mechanisms and the initialization procedures. The validation results on a simple test system illustrate that a high level of accuracy can be achieved using both approaches. In the particular test case under investigation, the RMSE values of all observed variables in both approaches are kept within 0.02 p.u., even for simulations of faults in the system. Future work will look to extend the analysis and the results to more realistic power system cases.

References

[1] "Common Grid Model Exchange Specification (CGMES)," ENTSO-E, Tech. Rep. Version 2.5, Draft IEC 61970-600 Part 1, Jul. 2016.

[2] F. J. Gómez, L. Vanfretti, and S. H. Olsen, "Binding CIM and modelica for consistent power system dynamic model exchange and simulation," in *2015 IEEE Power Energy Society General Meeting*, Jul. 2015, pp. 1–5.

[3] P. Palensky, A. A. v. d. Meer, C. López, A. Joseph, and K. Pan, "Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems," *IEEE Industrial Electronics Magazine*, vol. 11, no. 2, pp. 6–21, Jun. 2017.

[4] H. Krishnappa, "Model validation and feasibility analysis of Modelica based dynamic simulations using OpenIPSL and CGMES," MSc. Thesis, Delft University of Technology, 2017.

[5] C. Ivanov, T. Saxton, J. Waight, M. Monti, and G. Robinson, "Prescription for interoperability: Power system challenges and requirements for interoperable solutions," *IEEE Power and Energy Magazine*, vol. 14, no. 1, pp. 30–39, Jan. 2016.

[6] J. Britton, P. Brown, J. Moseley, and M. Bunda, "Optimizing operations with CIM: Today's grid relies on network analysis (and a lot of data)," *IEEE Power and Energy Magazine*, vol. 14, no. 1, pp. 48–57, Jan. 2016.

[7] L. Vanfretti, W. Li, T. Bogodorova, and P. Panciatici, "Unambiguous power system dynamic modeling and simulation using Modelica tools," in *2013 IEEE Power Energy Society General Meeting*, Jul. 2013, pp. 1–5.

[8] L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad, "iTesla power systems library (iPSL): A Modelica library for phasor time-domain simulations," *SoftwareX*, vol. 5, pp. 84–88, 2016.

[9] E. Hairer and G. Wanner, "Stiff differential equations solved by Radau methods," *Journal of Computational and Applied Mathematics*, vol. 111, no. 1, pp. 93–111, Nov. 1999.

[10] R. Viruez, S. Machado, L. M. Zamarreño, G. León, F. Beaude, S. Petitrenaud, and J. Heyberger, "A tool to ease Modelica-based dynamic power system simulations," in *Proceedings of the 12th International Modelica Conference*, Prague, Czech Republic, May 2017.

[11] S. C. Mueller, H. Georg, J. J. Nutaro, E. Widl, Y. Deng, P. Palensky, M. U. Awais, M. Chenine, M. Kuch, M. Stifter, H. Lin, S. K. Shukla, C. Wietfeld, C. Rehtanz, C. Dufour, X. Wang, V. Dinavahi, M. O. Faruque, W. Meng, S. Liu, A. Monti, M. Ni, A. Davoudi, and A. Mehrizi-Sani, "Interfacing power system and ICT simulators: Challenges, state-of-the-art, and case studies," *IEEE Transactions on Smart Grid*, no. 99, pp. 1–1, 2016.

[12] V. Jalili-Marandi, V. Dinavahi, K. Strunz, J. A. Martinez, and A. Ramirez, "Interfacing techniques for transient stability and electromagnetic transient programs," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 2385–2395, Oct. 2009.

[13] C. D. López, A. A. v. d. Meer, M. Cvetković, and P. Palensky, "A variable-rate co-simulation environment for the dynamic analysis of multi-area power systems," in *2017 IEEE Manchester PowerTech*, Jun. 2017, pp. 1–6.

[14] M. Stifter, R. Schwalbe, F. Andrén, and T. Strasser, "Steady-state co-simulation with PowerFactory," in *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, May 2013, pp. 1–6.

[15] F. Gonzalez-Longatt and J. L. Rueda, *PowerFactory applications for power system analysis*. Springer, 2014.

[16] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," *IEEE Transactions on Industry Applications*, vol. 44, no. 4, pp. 1286–1294, Jul. 2008.