Multi-simulation environment for smart grid : co-simulation approach

Author(s): Palizban, Omid; Kauhaniemi, Kimmo

Title: Multi-simulation environment for smart grid : co-simulation approach

Year: 2020

Version: Final draft (post print, aam, accepted manuscript)

Please cite the original version:

Palizban, O. & Kauhaniemi, K. (2020). Multi-simulation environment for smart grid : co-simulation approach. In: 2019 2nd International Conference on Smart Grid and Renewable Energy (SGRE), Doha, Qatar, 2019, pp. 1-5. https://doi.org/10.1109/SGRE46976.2019.9020905

Multi-Simulation Environment for Smart Grid: Co-simulation Approach

Omid Palizban and Kimmo Kauhaniemi
School of Technology and Innovations, University of Vaasa, Vaasa, Finland

Abstract— **Wide area measurement and control methods should be employed in modern grids to ensure a more resilient power system. Digital communication technologies play a vital role in such systems, and delays in the communication system can lead to malfunctions in the power system. The integration of the power system and communication network is thus a requirement for implementing smart grids. This paper investigates the possibility of comprehensive integration and discusses the challenges to coupling power system with communication network using continuous and discrete simulation techniques, respectively. The frameworks, platforms, and possible interfaces for a high-performance co-simulation are also presented; the effects of time synchronization and method for obtaining the accrued result are investigated. Finally, existing co-simulation methods are summarized in a table.**

Keywords— Communication system, Co-simulation, Smart grid, Time synchronization, Power system.

I. Introduction

Simulation technology falls into two major categories: physical and computer-based, which computer-based simulations can execute complex networks and are more cost-effective [1]. For the purposes of modeling smart grids, simulators can be classified as power network and communications systems simulators and three different approaches are considered here: comprehensive simulation, hardware in the loop, and co-simulation. Power network simulations can be combined with communications system simulations in a common environment to give the comprehensive simulation approach [2]. The main concept of Hardware in the Loop (HIL) is to integrate software simulators with a real hardware component to be tested. Such systems are usually used in power system analysis to test system control and protection functions [3]. In [4] provides a comprehensive review of the tools for evaluating a smart grid using HIL. The co-simulation method involves integrating several simulators to capture the interdependence of a network or process. In general, co-simulation is an option for coupling multiple subsystems with different modeling environments to simulate a system. In smart grid, the co-simulation approach usually involves power grid simulator and communications system to analyze a network. The method analyzes each system—the power network and the communication network—with dedicated simulators; these execute their respective simulations through appropriately designed Run-Time Interfaces (RTI) that coordinate the simulation management [5]. A comprehensive analysis into the requirements for co-simulations is presented in [6]. A major challenge in co-simulation is finding a synchronization mechanism for the different categories of combined simulators. The objective of this paper is to provide a comprehensive investigation of co-simulation in smart grids and present the impact of time and time synchronization in this method in detail.

Correct simulation timing and timely exchange of data between simulators must be ensured during the co-simulation. The time-stepped, global event-driven, and master–slave approaches are the three main methods of synchronization in this method [7] and there are also two different approaches to simulation timing: time resolution and time ratio [8]. The time resolution method presents a challenge when different time steps are used in the in simulators. For instance, the time step in power grid simulators for electromagnetic processes is on the order of milliseconds, while in steady state and electromechanical processes the step may be few seconds or more. The time steps used in communications system simulations are different, ranging from ten of millisecond (e.g., latencies in LANs) to seconds (e.g., latencies in WANs). The relation between simulation time and real time is also problematic [8]. As co-stimulation needs to involve fast planning and decision making, while also simulating several models with different configurations, it is crucial to run the model in a sufficient real time [9]. In this regard, regulation and adjustment of the time step is thus necessary for the systems investigated here.

The paper is organized as follows: all frameworks, platforms, interfaces related to the co-simulation are discussed in Section 2. The impact of time management and time synchronization both are very critical issue in this topic and are discussed in Section 3. The paper lists all available related works in co-simulation in Section 4. Finally, the paper is concluded in Section 5.

II. Framework, Platforms and Interface

A co-simulation usually includes two or more power grid simulators or one communication with at least one power grid simulator. A co-simulation should support both the transmission and distribution network levels with communications systems. There are a number of different conceptual approaches to co-simulation in the smart grid , and [10] present some frameworks of this method. Two standards describe how to combine different simulators in co-simulation: IEEE 1516-2000 (*High Level Architecture,* HLA) and IEEE 1516-2010 (*HLA Evolved*). These standards define the terminology, rules, object models, and interfaces needed for a co-simulation framework [6].

A. Platforms

About co-simulation platform, it may be either real time or non-real time (offline), which in Offline co-simulation (non-real time) method, the two different simulators —the power grid and the communications system—are generally run separately. However, as mentioned before, the simulation's time step should synchronize between the two simulators. The communications system thus runs first and its output is saved in a file. The power grid simulator is then run and reads those communications results back from the file. This method is not very accurate due to the asynchronous operation of the two simulators, but it is easily set up. In real time platforms, an interface is needed to exchange data between the two simulators and synchronize them. The two simulators run synchronously using a protocol such as TCP or UDP to exchange data. The main advantage of this approach is than it is capable of executing large and complex systems. However, a real time simulator is required and the initial set-up can be time-consuming [11].

B. Interface

The aim of interface development is to achieve more scalability, more reuse, and better interoperability. To better understand the co-simulation interface, it is worth becoming familiar with the different simulation tools in the power grid and communications system.

1) Power system simulators

There are two major types of power system simulator: a) *Steady state models*: these are usually used to model power network planning, energy markets, optimization, and Demand Side Management (DSM). They model the stable state of the power network. The simulation tools used are MATLAB, DigSilent, Power world, OpenDSS, and gridLab-D. b) *Transient dynamic models*: here the power system is modeled on the circuit level using differential equations. To discretize the equations, the trapezoidal rule is applied—for instance, to model sampling and switching events. Simulation tools generally solve the system equations repeatedly in each time step to determine the numerical time domain solutions. This simulation method is very common in power system control and protection. The simulation tools used for the method include PSCAD, PSLF, DigSilent, and MATLAB [12].

2) Communication network simulation

The principle of simulating communications systems is the processing and transmission of the message in order to model the system as a sequence of discrete events in a logical timeline [12]. Discrete event-driven simulation is a good option for systems that are subject to change by discrete events. The discretization of time into small intervals for continuous time systems cannot be used for communications systems (discrete event system) due to the difficulty of selecting the time step. Indeed, if the time step is too small or too large, simulation time is wasted, because the system state may not change for many consecutive time step or because many event can be missed in a single time step. As simulation tools used for such communications system, OPNET, NS-2, and NS-3 can be named.

III. Time Management and Synchronization

As mentioned earlier, in a co-simulation environment it is important to ensure a coordinated run time for different simulators (e.g., continuous simulation, fixed step-size simulation, variable step-size simulation, or event-driven simulation). To execute a co-simulation between multiple simulators, a scheduler is needed to keep track of time. At the start of the simulation, all simulators are at their initial times ($t = 0$). When asked to execute the next step of the simulation from simulators, the time coordinator (scheduler) is sent the time as initial time (t) and the simulators then send back the time for performing the next step ($t+1$). This time should be queued through the co-simulation time coordinator. The simulator step size is thus required to be constant; however, it can differ during the simulation. Figure 1.a describes the computation through simulation during the interval (t, $t+1$). If the two simulators *Sim I* and *Sim II* have the same step-size or resolution, the time step started by simulator *I* provides the all necessary input for the simulator *II*. The progress is shown in Figure 1.b. If the step size for *Sim II* is greater than that for *Sim I*, simulator *II* cannot use all the data provided by *Sim I*

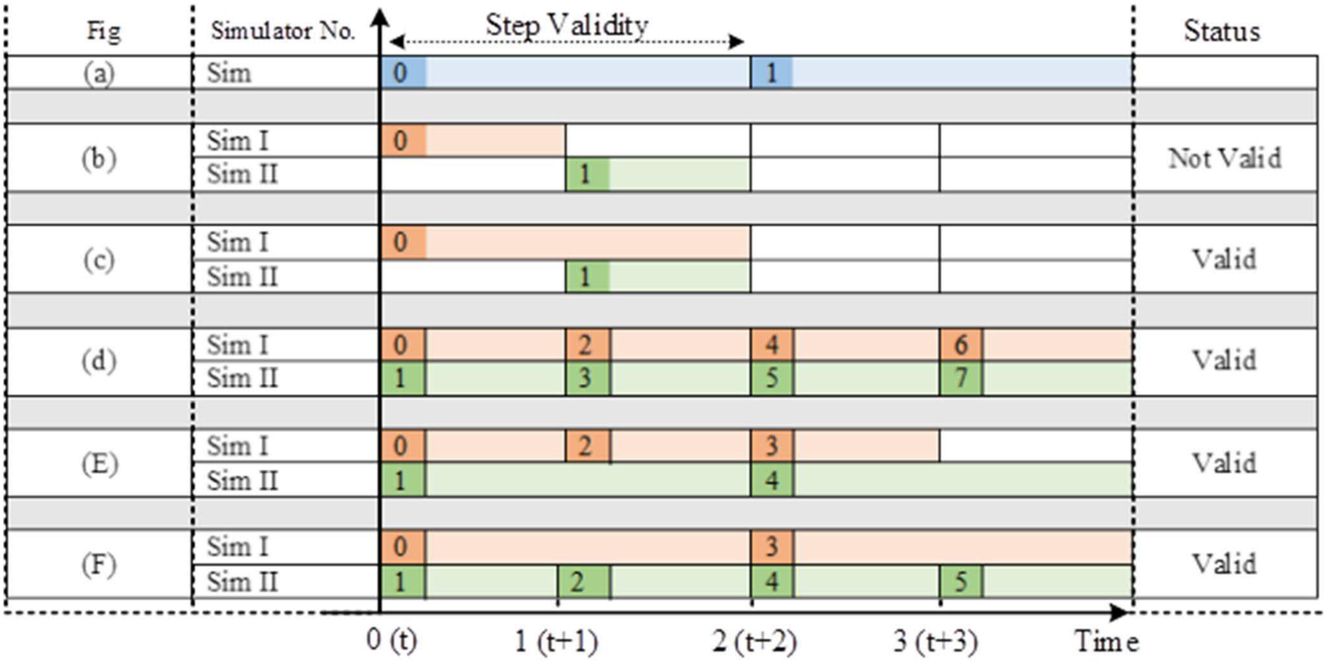

Fig.1 Schematic execution of two simulators

as *Sim II* can only use the input data that is valid at t (*Sim II*) [6]. In contrast, if the step size for simulator I is larger than that for II, simulator II uses similar data for several time (as long as it valid). These four scenario are depicted in Figures 1.C-F. [6]

As described, co-simulation requires synchronization mechanisms to ensure that data is transferred in the correct timestamp order between simulators. Time synchronization occurs automatically in real time co-simulations, and the major challenge in such systems is to ensure that all simulations are solved with the same data exchange time step and with communication between the simulators. However, in non-real time co-simulation, time synchronization is needed to ensure that the input values are available for all simulators when required [13].

Extant time synchronization methods fall into three types: *master-slave*, *time-stepped*, and *event-driven*.[12]

a) *Master-Slave*: in this approach, one simulation tool operates as a master simulator and the other simulators act as slaves to execute the model and exchange data. The precise logical time for interfacing is maintained only in the master simulator. This technique is suitable for steady state co-simulation with the power system simulator as slave and the communications simulator as the master. It can be used for models in which the slave simulators do not generate events or affect the master simulator.

b) *Time-stepped*: here all simulators execute separately until a particular logical time has been reached. The accumulated messages can be transferred between simulators when all of them have reached the specified logical time. One challenge to this method is errors that can occur due to time step delays between synchronization points; this can be solved by reducing the step size statically. However, the issue cannot be completely eliminated, even if the step size is altered adaptively.

c) *Event-driven*: in this synchronization technique, a message is sent and received randomly by a simulator. A time band during the simulation is assigned to each simulator to limit the execution of events in order to assure that the local and external events and messages are processed in a correct order. In this technique, the synchronization protocol can run in parallel or sequentially, depending on the implementation.

Reference [10] presents three strategies for synchronization: *conservative*, *speculative failover*, and *speculative recompute*. Although these provide on-time and consistent message delivery, their levels of resource usage and their performance characteristics differ. It is possible for users to select the strategy that best suits the availability of resources on the target computer system. The rest of this section discusses these three strategies in detail.

IV. Related Work

Different concepts of co-simulation and how to implement it have recently been described. In this section and in Table 1, a general summary is given of related works and of some special co-simulation methods between the power network, communications system, and control, protection, and market are discussed.

The Global Event-Driven Co-simulation Framework (GECO) [14] is a co-simulation framework that interfaces between PSLF and NS-2 for power system networks and communications system simulators, respectively. The time synchronization in this framework involves a global scheduler and global event queue. The main target of the work to analyze wide area measurements to protect and control PMU-based systems, while maintaining the cyber security of smart grid applications.

Co-simulation including the power system and communication network, as well as the protection and control system, in a discrete-event based manner is proposed in [15] as the *Toolkit for Hybrid Modelling of Electrical* power systems (THYME). The simulator is a publically available module of *A Discrete EVent System* simulator (ADEVS), which is based on the Discrete Event System Specification (DEVS). To implement a co-simulation system, this can be interfaced with NS2 and OMNeT++ as a communication network simulator. Another co-simulation project involves interfacing the Virtual Test Bed (VTB) as a simulator for power and control system, with OPNET Modeller as communication system simulators, with central coordinators for time synchronization (VEPNET) [16]. Interfacing Modelica and NS-2 (PowerNet) [17] and interfacing PSCAD and OPNET (Greenbench) [18] are other approaches to offline co-simulation; however, there have also been some published descriptions of real-time co-simulation methods. For instance, interfacing OPAL_RT with OPNET/SITL for the real time and hardware in loop simulations is widely used for power and control applications. [6]

A. Co-simulation of power systems, communications, and controls

In the modern grid, implementing an appropriate control system is one of the critical challenges. The communication systems have a vital role in controlling different distribution generators, energy storage system, and other components. In other words, characteristics of the communication channel (e.g., bandwidth, delay time,) can affect system performance. For instance, if measurement and monitoring of the power system are not possible due to a malfunction of the communication system or because the transfer of data between units is delayed, the controller will not be able to respond correctly, as it has not received accurate data. For studying of this kind of situation, co-simulation is a very useful way to interface power systems and communication networks.

Regarding related work on co-simulating power systems, communications, and controls, in [19] it is presented a coupling between continuous simulation of the power system and discrete event-based simulation of communications, which combines automation and control. A co-simulation of the power system with a voltage controller, considering the underlying communication network model, is presented in [20]. A co-simulation environment based on OMNeT++ (communication network) and OpenDSS (power system), involving integration between different distributed generators is described in [21]. Wireless charging for electric vehicles and interfacing with OMNeT++ as a communication infrastructure have also been investigated, with the aim of comparing different charging optimization algorithms. One possible way to implement and describe the control algorithm for IED utilizing the IEC61850 interoperability specification is to apply the IEC 61499 standard, recommended in the IEC Smart Grid roadmap. A control algorithm can act as a function

Table.1 Summary of related works to Co-simulation

Name	Objective	Power simulators	Communication Simulators	Computation	Synchronization	Scalability
GECO	Dynamic simulation for WAMPAC application	PSLF (Transient)	Ns-2	Ns-2	Event driven	Suitable for large system
TASSCS	SCADA cyber security, system virtualization	Power world	OPNET	OPENT	N/A (Static)	Suitable for large system
EPOCHS	Dynamic simulation for WAMPAC	PSCAD& PSLF (Transient)	Ns-2	External process	Time-Stepped	Suitable for large system
MAPNET	Dynamic simulation for WAMPAC	MATLAB (Transient)	OPNET	MATLAB	Master-Slave	Limited for single or small devices
VPNET	Distributed Control	Virtual test bed (Transient)	OPNET	VTB	Time-stepped	Limited for single or small devices
PowerNet	Network control	Modelica (Transient)	Ns-2	Ns-2	Time-stepped	Limited for single or small devices
INSPIRE	WAMPAC, IEC 61850	DIgSILENT (Transient)	OPNET	External process	Event driven (parallel)	Suitable for large system
Hybrid simulator [25]	WAMPAC	Adevs (steady state)	Ns-2	Ns-2	Master-Slave	Limited, have a rewrite code for different systems
Integrated simulation [2]	demand side management	Simulink (Steady state)	OMNET++	OMNET++	Master-Slave	Limited for small system
Combining simulation [26]	WAMPAC	Adevs-THYMS (Transient)	OMNET++	Adevs,OMNET++	Master-Slave	Suitable for large system
PC_Co-silmulation [27]	demand side management	OpenDSS (Steady state)	OMNET++	OMNET++	Master-Slave	Limited for small system
DMS cyber physical simulation [28]	demand side management	OpenDSS (Steady state)	Ns-2	MATLAB	Event driven	Limited for small system
Agent-based Co-simulation [29]	Protection and control	-	OPNET	Extended JADE	Event driven (parallel)	Limited for small system
IoT Co-simulation [30]	demand side management	GridLAB-D (Steady state)	CORE	GridLAB-D	Real time	Suitable for large system
Co-simulation platform [5]	Distribution Protection and control	OpenDSS (Steady state)	Ns-3	MATLAB	Event driven	Limited for small system
Co-Simulation of Distributed Smart Grid [12]	Agent based protection and control	PSCAD (Transient)	OPNET	Decomposi-tion	Event driven (parallel)	Limited for small system

block consisting of several different devices. The main feature of the standard (IEC 61499) is the event-based execution of the function blocks. In [22], an open source-based 4DIAC environment is used to implement the control algorithm in IEC 61499 control function blocks.

B. *Co-simulation of power system and protection*

The Integrated Co-simulation of Power and ICT systems for Real time Evaluation (INSPIRE) developed a HLA standard (IEEE 1516-2010) using DIgSILENT Power Factory and OPNET-Modeller as power network and communications system (ICT domain) simulators, respectively. The main objective of co-simulation here is to evaluate the reliability and performance of ICT infrastructures and protocols for the power system. ICT-based protection and control in a smart grid have been investigated in [23]. IEC 61850 and IEEEC37.118 were considered in INSPIRE, designed with generic architecture capable of integrating with applications and simulators implemented in C++, Java, and MATLAB. In [24] it is proposed an Electrical Power and Communication Synchronizing Simulator (EPOCHS), a co-simulator interfacing between power network and ICT simulations. This work is also based on HLA (IEEE 1516-2000) and utilizes PSCAD/EMTDC and PSLF as power system simulators and communication system, supported by NS-2 simulator. This approach is also used for studying agent-based protection and control in smart grid systems. A time-stepped synchronization, based on HLA time management, is used for both the EPOCHS and INSPIRE co-simulation frameworks.

V. CONCLUSION

Transferring the data required for controlling and sensing the power system through the communication network is the initial step in the vision of the modern grid. There are several different simulation tools for analyzing the power and communication systems separately, but it is necessary to combine these tools: co-simulation is thus needed for accurate investigation of the smart grid. This paper thus describes co-simulation and investigates the frameworks, platforms, and means of setting up an interface between these tools. The major challenges to implementing a co-simulation are time synchronization and the effect of the time management. Hence, three different methods (master-slave, time-stepped, and event-driven) and three different strategies (conservative, speculative failover, and speculative recompute) are presented for time synchronization. Moreover, the effects of time management are discussed on the example of two different systems. Some research into high-performance co-simulation techniques has already been done; this is summarized in a table that shows objectives, type of power and communication simulators, computation, synchronization methods, and scalability. In conclusion, more complex communication systems, different communication infrastructure architectures, the effect of the different communication protocol, and the performance of control power systems can be considered as topics for future work.

References

[1] A. Razaq, B. Pranggono, and H. Tianfield, “Simulating Smart Grid : Co-Simulation of Power and Communication Network,” no. September, pp. 1–4, 2015.

[2] K. Mets, T. Verschueren, C. Develder, E. Kevinmetsintecugentbe, T. L. Vandoorn, and L. Vandevelde, “Integrated Simulation of Power and Communication Networks for Smart Grid Applications,” *2011 IEEE 16th Int. Work. Comput. Aided Model. Des. Commun. Links Networks*, pp. 61–65, 2011.

[3] M. H. Cintuglu, T. Ma, and O. A. Mohammed, “Protection of Autonomous Microgrids Using Agent-Based Distributed Communication,” *IEEE Trans. Power Deliv.*, vol. 32, no. 1, pp. 351–360, 2017.

[4] C. Studies *et al.*, “Interfacing Power System and ICT Simulators :,” vol. 9, no. 1, pp. 14–24, 2018.

[5] M. Garau, G. Celli, E. Ghiani, F. Pilo, and S. Corti, “E valuation of S mart G rid C ommunication T echnologies with a C o -S imulation P latform,” *IEEE Wirel. Commun.*, vol. 24, no. April, pp. 42–49, 2017.

[6] C. Rehtanz and X. Guillaud, “Real-Time and Co-Simulations for the Development of Power System Monitoring , Control and Protection,” *2016 Power Syst. Comput. Conf.*, pp. 1–20.

[7] W. Li *et al.*, “Cosimulation for Smart Grid Communications,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2374–2384, 2014.

[8] F. Schloegl, S. Rohjans, S. Lehnhoff, J. Velasquez, C. Steinbrink, and P. Palensky, “Towards a Classification Scheme for Co-Simulation Approaches in Energy Systems,” pp. 1–6, 2015.

[9] M. Mirz *et al.*, “A Cosimulation Architecture for Power System , Communication , and Market in the Smart Grid,” vol. 2018, 2018.

[10] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal, “FNCS : A Framework for Power System and Communication Networks.”

[11] D. Bian *et al.*, “Real-time Co-simulation Platform using OPAL-RT and OPNET for Analyzing Smart Grid Performance,” pp. 1–5, 2015.

[12] C. Shum, S. Member, W. Lau, and S. Member, “Co-Simulation of Distributed Smart Grid Software Using Direct-Execution Simulation,” vol. 6, pp. 20531–20544, 2018.

[13] K. Pan, C. D. L, R. Bhandia, and P. Palensky, “Co-simulation Aspects for Energy Systems with High Penetration of Distributed Energy Resources.”

[14] H. Lin *et al.*, “GECO : Global Event-Driven Co-Simulation Framework for Interconnected Power System and Communication Network,” vol. 3, no. 3, pp. 1444–1456, 2012.

[15] “THYME: Toolkit for HYbrid Modeling of Electric power systems.” [Online]. Available: http://web.ornl.gov/~1qn/thyme/docs/.

[16] W. Li, A. Monti, M. Luo, and R. A. Dougal, “VPNET : A Co-simulation Framework for Analyzing Communication Channel Effects on Power Systems,” pp. 143–149, 2011.

[17] V. Liberatore, I. Computer, and A. Al-hammouri, “Smart Grid Communication and Co-Simulation,” no. 1, pp. 1–5, 2011.

[18] M. Wei and W. Wang, “Greenbench : A Benchmark for Observing Power Grid Vulnerability Under Data-Centric Threats,” *IEEE INFOCOM 2014 - IEEE Conf. Comput. Commun.*, pp. 2625–2633, 2014.

[19] M. Stifter, J. H. Kazmi, and F. Andr, “Co-Simulation of Power Systems , Communication and Controls,” 2014.

[20] R. Mosshammer, F. Kupzog, M. Faschang, and M. Stifter, “Loose Coupling Architecture for Co-Simulation of Heterogeneous Components,” *IECON 2013 - 39th Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 7570–7575, 2013.

[21] L. Martin, D. Qian, G. J, and M. Maier, “Co-Simulation of PEV Coordination Schemes over a FiWi Smart Grid Communications Infrastructure dl : Il :,” pp. 2901–2906, 2012.

[22] A. Zoitl, T. Strasser, and A. Valentini, “Open Source Initiatives as basis for the Establish- ment of new Technologies in Industrial Automa- tion : 4DIAC a Case Study,” *2010 IEEE Int. Symp. Ind. Electron.*, pp. 3817–3819, 2010.

[23] H. Georg *et al.*, “Analyzing Cyber-Physical Energy Systems : The INSPIRE Cosimulation of Power and ICT Systems Using HLA,” vol. 10, no. 4, pp. 2364–2373, 2014.

[24] K. Hopkinson *et al.*, “EPOCHS : A Platform for Agent-Based Electric Power and Communication Simulation Built From Commercial Off-the-Shelf Components,” vol. 21, no. 2, pp. 548–558, 2006.

[25] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, “Integrated Hybrid-Simulation of Electric Power and Communications Systems,” 2007.

[26] J. Nutaro, “Designing power system simulators for the smart grid : combining controls , communications , and electro-mechanical dynamics,” *2011 IEEE Power Energy Soc. Gen. Meet.*, pp. 1–5, 2011.

[27] M. Lévesque and M. Maier, “Communications and Power Distribution Network Co-Simulation for Multidisciplinary Smart Grid Experimentations,” no. 413427, 2011.

[28] G. Celli, P. A. Pegoraro, F. Pilo, S. Member, G. Pisano, and S. Sulis, “DMS Cyber-Physical Simulation for Assessing the Impact of State Estimation and Communication Media in Smart Grid Operation,” vol. 29, no. 5, pp. 2436–2446, 2014.

[29] F. Perkonigg, D. Brujic, and M. Ristic, “Platform for Multiagent Application Development Incorporating Accurate Communications Modeling,” *IEEE Trans. Ind. Informatics*, vol. 11, no. 3, pp. 728–736, 2015.

[30] X. I. N. Li, Q. Huang, and D. Wu, “Distributed Large-Scale Co-Simulation for IoT-Aided Smart Grid Control,” pp. 19951–19960, 2017.