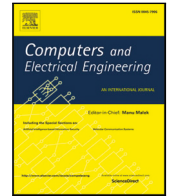


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Co-simulation platform for interconnected power systems and communication networks based on PSS/E and OMNeT++[☆]

Pablo Ledesma^{*}, Davide Gotti, Hortensia Amaris

Universidad Carlos III de Madrid, Av. Universidad 30, 28911 Leganés, Madrid, Spain

ARTICLE INFO

Keywords:

Dynamic simulation
Co-simulation
Transient stability
Transmission network
OMNeT++
PSSE

ABSTRACT

This paper proposes a co-simulator that combines OMNeT++ for communication systems with PSS/E for the electrical transmission network. The co-simulator applies an event-driven synchronization method that minimizes errors due to delays in the synchronization between both simulators. The synchronization method pauses the simulation of the power system at each communication event, while a supervisory module in PSS/E returns control to the event simulator if any condition from a pre-specified set is met. The proposed co-simulator is demonstrated on a protection system based on peer-to-peer communication and used to evaluate the effect of communication latency times on an online state estimator.

1. Introduction

Integration of information and communications technology (ICT) is increasingly relevant in power systems. ICTs affect many aspects of the operation and control of transmission systems such as monitoring, protection, demand response and wide-area control of power electronic resources. ICT systems add functionality to power systems and can improve their reliability and flexibility, but they can also contribute to failures if not thoroughly tested and they bring the threat of cyberattacks. It is therefore necessary to model the combined operation of power and ICT systems to assess its performance during normal operation and under abnormal conditions such as severe disturbances.

The main difficulties of simulating in a single platform the electrical network and communications technology arise from the different nature of both systems. On the one hand, mathematical models representing the dynamics of electrical networks include ordinary differential equations (ODEs) that are discretized and integrated with a constant or variable time step. On the other hand, communication systems are event-driven systems in which ODEs play no part. Computer simulators that integrate both systems must define a synchronization procedure that allows the coordinated progress of the numerical integration time and the event-based time [1].

The problem of interfacing power system and ICT simulators have been extensively studied during the last decade [2,3]. Most proposed solutions are centered on distribution systems, applying smart grid concepts to distributed generation, aggregated loads or microgrids. These solutions combine an event-driven simulator for communications with a power system simulator centered on distribution systems such as EPRI's OpenDSS [4–8], or with a power system simulators intended for electromagnetic transients [9,10]. It is however difficult to extend these solutions designed for distribution networks to the study of transmission systems for two reasons. Firstly, studies typically performed in transmission systems such as frequency control and rotor angle stability need extensive libraries with models of power plant controls (e.g. turbine governors, excitation systems and power system

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Mohamed Elhabib Maicha.

^{*} Corresponding author.

E-mail addresses: pablole@ing.uc3m.es (P. Ledesma), dgotti@ing.uc3m.es (D. Gotti), hamaris@ing.uc3m.es (H. Amaris).

<https://doi.org/10.1016/j.compeleceng.2022.108092>

Received 1 September 2021; Received in revised form 6 May 2022; Accepted 11 May 2022

Available online 27 May 2022

0045-7906/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

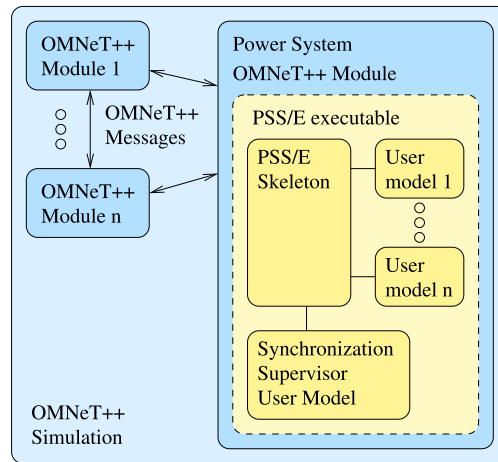


Fig. 1. Overall structure of the co-simulator.

stabilizers) and protections. These model libraries are currently found only in specialized software tools. Secondly, the sheer scale of modern interconnected systems, that may include hundreds of power plants and thousands of busbars, requires a dedicated, optimized software to solve the power flow equations and the numerical integration of the dynamic models. When power system monitoring, control and security assessment rely on synchrophasors [11], the problem of the computational load is aggravated by the volume of data that must be collected and sent across the system.

There is a number of open source tools to model transmission systems, such as MATPOWER/MatDyn, PYPOWER-Dynamics and GridPack [12]. However, currently these projects are small in scale and their model libraries are limited. Two of the main software tools used by the industry to model large-scale transmission systems are PowerFactory by DgSILENT and PSSE by Siemens. While there has been a number of works on co-simulation platforms that use PowerFactory [13–15], there is a lack of alternatives using PSS/E in spite of it being one of the tools most extensively used by transmission system operators (TSOs). Recent versions of PSS/E can be run from a Python interpreter, which facilitates embedding it in other platforms written for example in Python or C++. This opens new possibilities of interaction between PSS/E and event-based simulators.

The main contribution of this paper is a co-simulation platform based on OMNeT++ and PSS/E that coordinates the time-domain simulation of power networks with the event-base simulation of communication networks. The proposed method gives access to the extensive dynamic model library contained in PSS/E and makes it possible to take advantage of user models for specific components and power system cases already developed by TSOs. The platform is compared with other similar works and its performance is demonstrated by showing two basic applications of the co-simulation of transmission and communication systems: a back-up protection system based on peer-to-peer communication and an online state estimator based on an unscented Kalman filter (UKF). Co-simulation studies of communication and dynamic state estimators in transmission systems are scarce [16], and constitute a second contribution of this work.

The structure of the paper is as follows: Section 2 describes the general method adopted in this work; Section 3 puts it in context with other co-simulation studies; Sections 4 and 5 demonstrate the proposed method on two case studies; and Section 6 is the conclusion.

2. Co-simulation framework

2.1. General approach

The simulation of the electrical network and the communication system must be arranged in such a way that they advance together and exchange information at critical points that, generally speaking, cannot be known in advance. To achieve this goal, the proposed co-simulator implements a master–slave architecture, represented in Fig. 1, in which the OMNeT++ simulator works as master and PSS/E as slave. The simulation of the power system is embedded in the Power System OMNeT++ Module that exchanges messages with other modules representing components of the communication system.

The PSS/E skeleton in Fig. 1 contains the model that represents the electrical network together with the most common power system components (synchronous machines, turbine governors, excitation systems, power system stabilizers, etc.). In addition to this basic structure, it is possible to write user models and integrate it with the skeleton. User models are primarily conceived to represent other dynamics present in the power system, but they can also be used for other purposes. In the proposed co-simulator, a new model called Synchronization Supervisor User Model (SSUM) takes charge of the synchronization between both simulators, applying the method described in Section 2.4.

The Power System OMNeT++ Module runs together with other OMNeT++ modules representing relays, protection systems, centralized controls, etc. The Power System module is responsible for running the PSS/E simulator and for interacting with any other OMNeT++ modules that need to exchange information with it. For example, an OMNeT++ module representing a relay may send a message to the Power System Module asking for the measurement of a certain voltage or current. Another example is an OMNeT++ model representing a protection sending a message to the Power System Module with an order to open a certain automatic breaker to clear a fault.

2.2. Dynamic simulation of the power network

Modern interconnected power systems are represented by complex, nonlinear, discrete–continuous dynamical mathematical models. The equations that define the dynamic behavior of a power system can be formulated as [17]:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}, \boldsymbol{\eta}, \mathbf{u}, t) \quad (1)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{y}, \boldsymbol{\eta}, \mathbf{u}, t), \quad (2)$$

where \mathbf{x} ($\mathbf{x} \in \mathbb{R}^{n_x}$) is the vector of state variables, for example the angle and speed of the synchronous generators; \mathbf{y} ($\mathbf{y} \in \mathbb{R}^{n_y}$) is the vector of algebraic variables, for example the magnitude and angle of the bus voltages; $\boldsymbol{\eta}$ ($\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$) are the controllable parameters, for example references of voltage and power in power plants; \mathbf{u} ($\mathbf{u} \in \mathbb{R}^{n_u}$) are the discrete variables and t represents the time. Function $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_\eta} \times \mathbb{R}^{n_u} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{n_x}$ represents the differential equations, and $\mathbf{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_\eta} \times \mathbb{R}^{n_u} \times \mathbb{R}^+ \rightarrow \mathbb{R}^{n_y}$ represents the algebraic equations. Generally speaking the transmission grid is represented by algebraic equations, while synchronous and non-synchronous generation, HVDC links, FACTS, control systems and other components whose dynamics are simulated are represented by differential equations. Discrete variables are used to represent discrete controls, protection relays and external disturbances.

This work utilizes PSS/E as the simulation tool to implement and solve the model represented by Eqs. (1) and (2). PSS/E is a transmission planning and analysis software used across the globe by many TSOs. It has been used in this work to simulate the electrical transmission network for two reasons: firstly, it contains an extensive model library of power system components such conventional power plants, control systems, non-synchronous power parks, HVDC lines, FACTS, load models and protection systems; secondly, it offers the possibility of using the power system models already developed and maintained by TSOs.

In addition to the standard model library, users can program their own models and include them in the simulation. User models must be developed in Fortran, compiled prior to the simulation and linked with the PSS/E skeleton. User models are usually designed to represent physical devices, but can also be used to retrieve run-time information and control the progress of the simulation.

Simulations in PSS/E perform a numerical integration of the ODEs that define the dynamics of the power system. PSS/E, like all power system dynamic simulators, is discrete time-driven, as opposed to communication simulators that are event-driven. In order to solve Eq. (1), PSS/E applies the explicit, two-step Adams–Bashforth method [18,19]:

$$\begin{aligned} \mathbf{x}_{n+1} = & \mathbf{x}_n + \frac{3}{2} \Delta t \mathbf{f}(\mathbf{x}_n, \mathbf{y}_n, \boldsymbol{\eta}, \mathbf{u}_n, t_n) \\ & - \frac{1}{2} \Delta t \mathbf{f}(\mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \boldsymbol{\eta}, \mathbf{u}_{n-1}, t_{n-1}), \end{aligned} \quad (3)$$

where Δt is the time step and subindexes $n-1$, n and $n+1$ represent the value of the corresponding variables at times t_{n-1} , t_n and t_{n+1} , respectively.

One feature of PSS/E that is central to the proposed co-simulator is that it is possible to run any PSS/E activity from a Python interpreter, which makes it possible to launch an instance of PSS/E from Python and embed it in a C++ program.

2.3. Simulation of communication networks

Communication networks can be modeled as event-driven systems. Following the formalism Discrete Event System Specification (DEVS) [20], event systems can be defined as the following 7-tuple:

$$M = \langle X, Y, S, t_a, \delta_{ext}, \delta_{int}, \lambda \rangle, \quad (4)$$

where

X and Y are the set of input and output events, respectively;

S is the set of partial states;

$t_a : S \rightarrow \mathbb{T}^\infty$ is the time between the current event and the next event;

$\delta_{ext} : Q \times C \rightarrow S$ is the external transition function, where $Q = \{(s, t_e) | s \in S, t_e \in (\mathbb{T} \cap [0, ta(s)])\}$ is the set of total states, with t_e being the elapsed time since the last event;

$\delta_{int} : S \rightarrow \mathbb{T}^\infty$ is the internal transition function;

$\lambda : S \rightarrow Y$ is the output function.

In the case of a communication system, S is the state while a discrete event is being processed; t_a is the time between two consecutive events, δ_{ext} represents the effect of an input on the system state, and δ_{int} represents the internal change of a state once the lifetime of the state is reached [21].

The DEVS formulation can be also applied to the power system simulation, in which case S is the set of state variables, t_a is the time step Δt , and δ_{int} represents the change in the state variables after each integration step.

The co-simulation of the power system and the communication system can be described as a coupled DEVS defined by the following 7-tuple:

$$N = \langle X, Y, D, \{M_i | i \in D\}, C_{xx}, C_{xy}, C_{yy}, Select \rangle, \quad (5)$$

where

X and Y are the set of input and output events, respectively;

D is the set of sub-components, in our case $D = PS, COM$ where PS represent the power system and COM the communication network;

M_i is the set of atomic DEVS models contained in the coupled DEVS, in our case both the DEVS models for the power system and the communication network;

C_{xx} and C_{yy} are the set of external input couplings and the set of external output couplings, in our case they are both empty;

C_{xy} is the set of internal couplings, in our case $C_{xy} = (Y_p, X_c) \cup (Y_c, X_p)$;

$Select$ is the tie-breaker function in the case of simultaneous events.

The DEVS models defined in (4) and (5) are solved using OMNeT++ [22], which is a well-known event-driven simulator library and framework primarily designed for building network simulators. It is an open source, modular tool in which network components are modeled as separate modules written in C++ that exchange messages. In OMNeT++ the arrivals of new messages are the typical events that drive the simulation.

One of the main advantages of using OMNeT++ is the availability of external simulation models that cover a variety of protocols and technologies. Another important feature is that OMNeT++ facilitates the collection of data over multiple simulations to perform statistical analysis on the results.

2.4. Time synchronization

The coordination between the numerical integration of the ODEs defining the dynamics of the power system and the events produced by communication systems is a fundamental issue that all co-simulation platforms need to solve.

One simple method to coordinate time-domain integration and communication events is time-stepped synchronization, in which both simulators run independently and halt at fixed times to exchange information. This method is easy to implement and can be useful in certain cases when the frequency of the communication between both simulators is not critical [23]. However, time-stepped synchronization can cause significant and cumulative errors because the exchange of information between simulators must wait until the next synchronization point, and may consume excessive computational resources if the simulation of the power system is forced to stop too frequently to check for relevant events.

This work implements an event-driven synchronization scheme in which the need for synchronization is checked at every time step and at every communication event. Fig. 2 shows the event-driven synchronization process, with the marks on the PSS/E timeline representing the integration time step Δt . The default time step in a PSS/E dynamic simulation is one half of the period, this is 0.01 s at 50 Hz and 0.0083 s at 60 Hz, which is sufficient for the time scale of electromechanical oscillations in power systems [24].

OMNeT++ keeps a record of all future events in a list named List of Future Events (LFE). Every time an OMNeT++ module finishes processing an incoming message, it reads the time at which the next event in the LFE is scheduled and sends an order to the PSS/E simulator to continue its simulation until that time (Fig. 2(a)). Halting PSS/E at every OMNeT++ event ensures a perfect synchronization between both simulators, because each OMNeT++ event may trigger new future events at times that cannot be known in advance.

When PSS/E performs the simulation it checks a set of pause-simulation conditions at every iteration step. Two things may happen:

1. no condition is matched and PSS/E runs the simulation until the time of the next scheduled event (Fig. 2(b)), or
2. while PSS/E is running the simulation, one or more of the conditions is matched. Then PSS/E halts the simulation, inserts a new event and returns the control of the simulation to OMNeT++ (Fig. 2(c)).

This approach minimizes the errors because no delay is introduced in the exchange of information between both simulators.

2.5. Implementation

In the proposed architecture calls to PSS/E commands, or *activities* in the PSS/E terminology, are python commands embedded in the C++ code of the OMNeT++ Power System module. In every simulation there is one and only one Power System module that creates and maintains an instance of a PSS/E case. The rest of the OMNeT++ modules depend on the specific study carried out, and typically represent protections, measurement devices and control systems with their respective communication systems.

A new user model called the Synchronization Supervisor User Model (SSUM) is implemented as the part of the PSS/E executable, as shown in Fig. 1. The SSUM checks at every iteration step whether any condition that requires a pause in the PSS/E simulation is met. The SSUM is written in Fortran as a separate PSS/E user module, compiled before the simulation and linked with the main skeleton of PSS/E.

Fig. 3 shows the information exchanged between OMNeT++ and PSS/E within the Power System module:

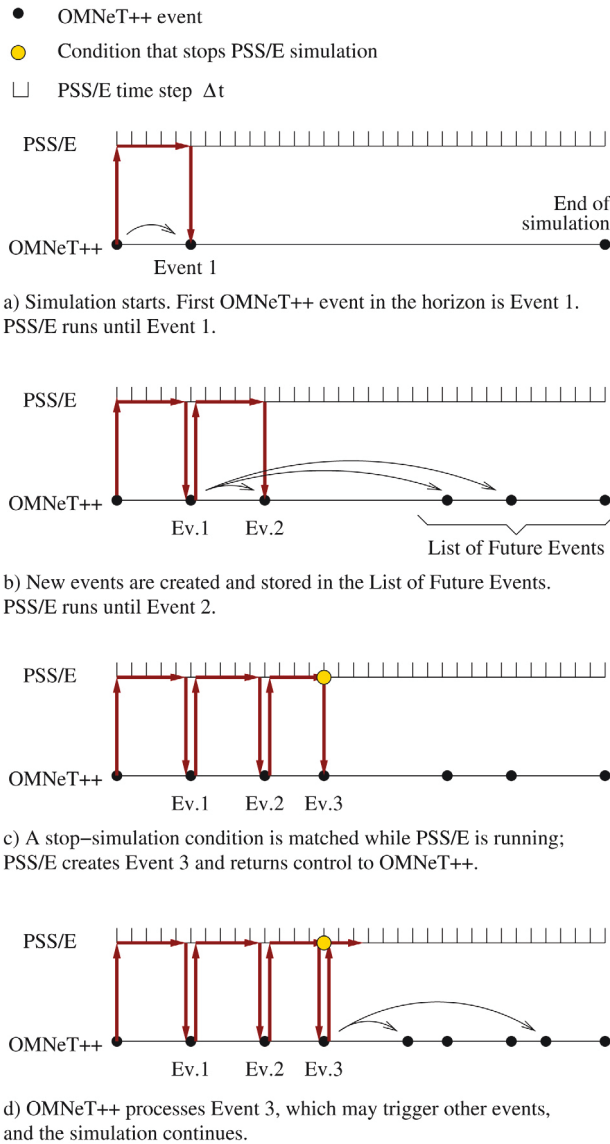


Fig. 2. Example of time advance in PSS/E and OMNeT++ simulators.

Initialization. OMNeT++ launches an instance of PSS/E and orders the loading of a case and the initialization of the simulation. The initialization includes selecting the variables to be monitored (PSS/E channels), calculating the initial conditions of all the state variables, and calculating the derivatives in the initial state to check that the simulation begins in a steady-state regime.

Run simulation. OMNeT++ orders PSS/E to run a simulation until the time at which the next event is scheduled.

Modify system. OMNeT++ orders a change in the PSS/E simulation. This change can be for example the setting of a fault, a switching operation or a change in a control parameter.

Read variable. OMNeT++ retrieves the value of a PSS/E variable, for example a bus voltage or a branch current. In typical studies electrical variables are needed by protections and measuring devices.

Pause information. When the SSUM pauses the numerical integration, the power system OMNeT++ module sends a message with information about the condition that has paused the simulation to all subscribed OMNeT++ modules. This condition can be, for example, a current value that exceeds a setting in a protection modeled in OMNeT++.

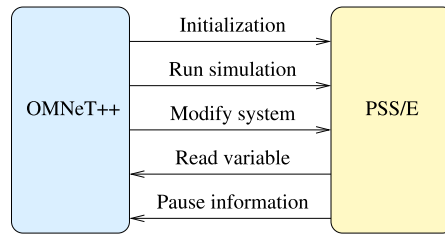


Fig. 3. Communication between OMNeT++ and PSS/E.

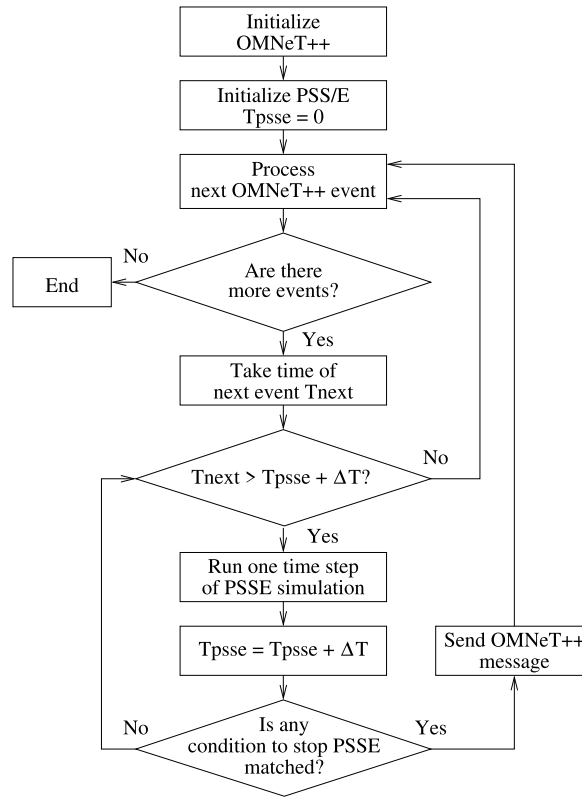


Fig. 4. Cosimulator flowchart.

Fig. 4 shows the general flowchart of the co-simulation platform. After the initialization of both simulators OMNeT++ processes its first event, which may trigger more events in the future, and reads the time of the next event in the List of Future Events. Whenever the next event is further down in the time horizon than the current simulation time in the power system, PSS/E is ordered to continue the simulation until it reaches the time of the next scheduled event. After each time step during the numerical integration, the SSUM in PSS/E checks the power system for conditions that must pause the PSS/E simulation. If such a condition is met, PSS/E pauses the simulation and returns control to OMNeT++.

The current PSS/E architecture imposes some constraints to the practical implementation of the proposed scheme: the co-simulator must be built on a Windows platform, a specific commercial Fortran compiler must be used to compile the SSUM, and a 32-bit version of OMNeT++ must be used to link it with the PSS/E dynamic libraries. Ref. [25] contains the basic set of files needed to execute an example of the proposed co-simulation platform.

3. Related works

Several co-simulation platforms of power systems and communications have been proposed, most of them during the last decade. This work shares some characteristics, and differs in others, with these platforms. Table 1 orders chronologically some of the most relevant co-simulators and summarizes their main characteristics. Given the difficulty of accessing the original data used to represent

Table 1
Co-simulation platforms.

Work	Year	Base case	Power system	Comm.	Synchronization	Scalability
EPOCHS [28]	2006	WAMPAC	PSCAD/PSLF	NS-2	Time-stepped	Large, Transmission
VPNET [29]	2011	Device control	VTB	OPNET	Time-stepped	Small
PowerNet [30]	2011	Device control	Modelica	NS-2	E-D	Small
ORNL [31]	2011	No base case	Adevs-THYME	OMNeT	Pause at iterations	No model library
GECO [21]	2012	WAMPAC	PSLF	NS-2	Pause at iterations	Large, Transmission
Levesque et al. [32]	2012	DSM	OpenDSS (NT)	OMNeT	–	Medium, Distribution
INSPIRE [33]	2014	WAMPAC	DigSILENT	OPNET	Distributed	Large, Transmission
Celli et al. [34]	2014	DSM	OpenDSS (NT)	NS-2	E-D	Medium, Distribution
Greenbench [35]	2014	Cybersecurity	PSCAD	OMNeT	Pause at iterations	Small, Distribution
Li et al. [27]	2017	DSM	GridLAB-D (NT)	CORE	Real-time	Medium, Distribution
Garau et al. [36]	2017	DPC	OpenDSS (NT)	NS-3	E-D	Medium, Distribution
Shum et al. [10]	2018	DPC	PSCAD	OPNET	E-D	Small
Mana et al. [37]	2021	Microgrids	GridLAB-D	NS-3	E-D	Medium, Distribution
Deng et al. [38]	2021	WAMPAC	PSS/E and PSLF	–	E-D	Large, Transmission
This work	2021	WAMPAC	PSS/E	OMNeT	E-D with SSUM	Large, Transmission

WAMPAC: Wide Area Monitoring, Protection and Control; DSM: Distributed System Management; DPC: Distribution Protection and Control; NT: No Transients; E-D: Event-Driven.

power systems in previous works, this section focuses on two main aspects: firstly on the scope of application and scalability of the co-simulator, and secondly on the method of synchronization. Other comparisons between co-simulators of power and communication systems can be found in [21,26,27]

3.1. Scope and scalability

A distinction can be made between co-simulators centered on distribution systems and/or microgrids, and co-simulators centered on large interconnected transmission networks. The different characteristics of distribution and transmission networks determines the selection of the power system simulator. The size of large interconnected systems requires specialized mathematical methods to solve the electrical network and a wide range of dynamic models to represent components such as power plants and their controls. As a result the scope and the scalability of each co-simulation platform are closely related, because it is difficult to apply a simulator designed for distribution networks to the transmission level.

A number of co-simulators centered on distribution systems use open-source software OpenDSS with the aim of supporting the integration of distributed resources in smart grids, as is the case of [32,34,27,36]. These platforms cannot be extended to the dynamic analysis of power systems for lack of appropriate models and because OpenDSS performs time-driven loadflow analysis; this is, OpenDSS does not perform the numeric integration of the differential equations representing the dynamics of the power system. Li et al. [27] follow a similar approach but use GridLAB-D to simulate the distribution system. The time scale of the studies conducted in these works is frequently hours or even days, which makes the synchronization method less critical compared to transient stability studies.

Mana et al. [37] and Shum et al. [10] are also centered on the distribution level but use power system simulators that do perform dynamic simulations. This makes it possible to analyze the evolution of frequency and voltage in small distribution systems, although these co-simulators are still not applicable to transmission systems because GridLAB-D and PSCAD are not designed to represent large interconnected networks.

EPOCHS [28], ORNL [31], GECO [21], INSPIRE [33] and, more recently, Deng et al. [38] are able to simulate large interconnected systems. From a practical point of view the scalability of ORNL [31] is limited because its power system simulator, open source software THYME, lacks an extensive library model. The rest of these platforms use software tools extensively used by transmission system operators across the globe to represent large interconnected grids: PSLF by GE, DIGSiLENT by PowerFactory and PSS/E by Siemens. The proposed co-simulator shares this approach by using PSS/E, which guarantees the scalability of the simulator to handle large transmission systems.

3.2. Method of synchronization

Early co-simulators tend to use time-stepped synchronization, which has the problem of incurring in simulation errors as discussed in Section 2.4. Most recent co-simulators, however, apply event-driven synchronization techniques that pause the power system simulator when a relevant event occurs in the communication system.

Several works such as ORNL [31], GECO [21] and Greenbench [35] apply an event-driven synchronization in which each of the iterations of the power system dynamic simulation is treated as a discrete event, so that the list of future events contains all the power system iteration steps. This approach simplifies the synchronization but can be inefficient when communication events are scarce, as illustrated in Section 4.2.

PowerNet [30] applies a different type of event-driven synchronization in which the simulation only pauses at the times prescribed by the communication system simulator. It is not possible for the power system simulator to trigger a communication

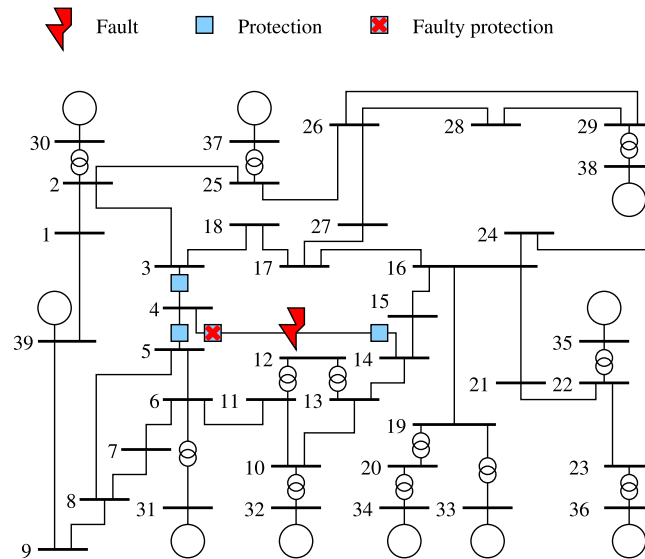


Fig. 5. IEEE 39 Bus test system.

event, for example if as a result of a fault an electrical variable crosses a threshold set in a relay, which can result in a loss of synchronization with the subsequent error.

INSPIRE [33] applies a synchronization method in which different simulators communicate with each other to determine the time frame in which it is safe to advance the simulation without risking the arrival of a new event. This approach is suitable for parallel computing and has the advantage of being easily adapted to new simulators, but it is not time-efficient and can produce errors when applied to simulators with different time steps.

The work by Deng et al. [38] has been included in this comparison because it integrates PSS/E in a co-simulator to perform dynamic studies in a large interconnected system. However, this work combines two power system simulators without any event-driven communication simulator, which makes synchronization easy if both power system simulators apply the same time step.

Compared to other co-simulators for large interconnected transmission systems this work applies an event-driven synchronization method that includes a supervisory module in the power system simulator. This supervisor, here called the SSUM, pauses the simulation when it detects a relevant event such as a violation of a voltage or current limit in a protective relay, as described in Sections 2.4 and 2.5. This approach minimizes errors due to synchronization while maximizing the computation speed during periods of scarce communication events.

4. Case study 1: Communication-based distance relay protection

4.1. Description

The aim of this study is to demonstrate the performance of the co-simulation platform in a case that depends on peer-to-peer communication between remote agents. In order to provide a term of comparison with a previous study, the case is based on a distributed protection system described in detail in [21]. The protection system consists of main and backup protections that communicate with each other without the intervention of a central control system. When a protection detects a fault in its own zone it trips immediately; when a protection detects a fault in a zone in which it is acting as a backup, it sends a query to the relays responsible for that zone and makes a decision based on the response. If the faults persists after 80 ms the backup protection trips. The breaker opening time is 60 ms.

The protection system is applied to the IEEE 39 Bus Test Case shown in Fig. 5. It consists of two distance relays protecting Line 4–14 at both extremes, and two distance relays connected in lines 3–4 and 4–5 that act as communication-based backup protections. A 3-phase short-circuit is applied in the middle of Line 4–14. The protection at bus 14 detects the fault and opens the line but the protection at bus 4 reports a failure, causing the back-up protections at substations 3 and 5 to trip.

During the simulation, OMNeT++ modules representing the relays subscribe to messages from the power system OMNeT++ Module. The SSUM in PSS/E monitors the impedance seen by the different relays, and when the impedance seen by a relay crosses its setting point the SSUM pauses the PSS/E simulation and the power system OMNeT++ module sends a message to the corresponding relays with the value of their fault impedances. Upon receiving this message each relay acts according to its fault impedance, which may correspond to a fault in the zone where it is acting as the main protection or to a zone where it is acting as a backup protection.

Communication links between protections in different substations are modeled with a 10 ms time delay. The time step in PSS/E is set at 0.01 s. For the purpose of simplicity, no background traffic is considered.

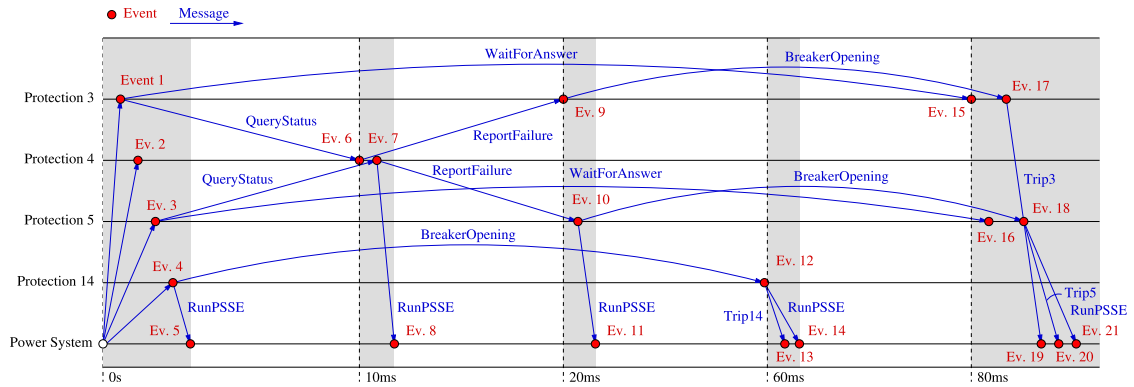


Fig. 6. OMNeT++ sequence chart.

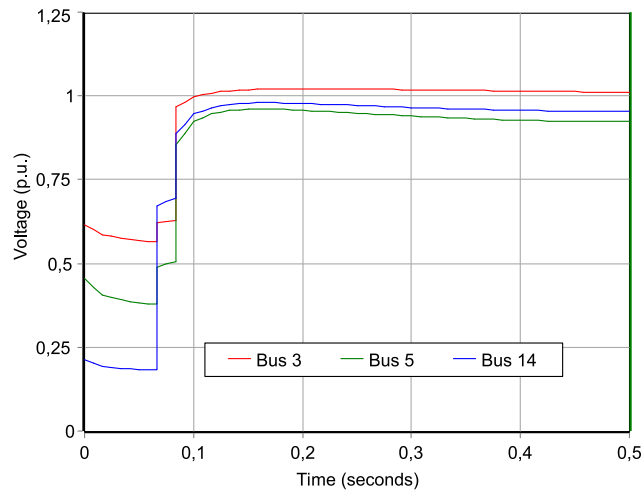


Fig. 7. Bus voltages.

4.2. Results

Fig. 6 shows the sequence chart with the messages and events of the simulation. The fault is applied at time $t = 0$. As soon as the fault is applied, the SSUM detects fault impedances beyond the set points of the protections and sends messages to protections 3, 4, 5 and 14 with their corresponding fault impedances. The modules of the back-up protections at buses 3 and 5 send a QueryStatus message to the module of the protection at bus 4 asking for its current status, receive a failure message and consequently open their circuit breakers. The simulation of the power system in PSS/E is performed in 5 parts, pausing at 10, 20, 60 and 80 ms before continuing to the end of the simulation.

Fig. 7 shows the voltage at substations 3, 5 and 14. The circuit breaker at bus 14 opens 60 ms after the fault and causes the voltage at bus 14 to recover from 0.2 to 0.7 p.u. approximately, while the circuit breakers at buses 3 and 5 open 80 ms after the fault occurs.

Fig. 8 shows the speed deviations of the 10 synchronous generators in the system during the first 6 s. The system remains stable after the fault and the frequency increases by 1.5% approximately.

The synchronization method between OMNeT++ and PSS/E proposed in this work eliminates cumulative errors due to synchronization by pausing the PSS/E simulation between OMNeT++ events. This method ensures that the power system simulations pauses whenever the communication system needs to be updated, but it may also increase the wall-clock simulation time compared to other time-stepped or event-driven synchronization methods. The efficiency of the proposed method depends on the frequency of communication events compared to the time-scale of electromechanical oscillations in the power system. If there are many communication events per PSS/E time step and they trigger complex calculations, most of the CPU time is consumed by the simulation of the communication system and the synchronization with PSS/E does not increase significantly the computational effort. Conversely, if there are few communication events then the PSS/E simulation runs uninterrupted for long periods of time and the synchronization process does not increase the computation time either. The least efficient case corresponds to a frequency

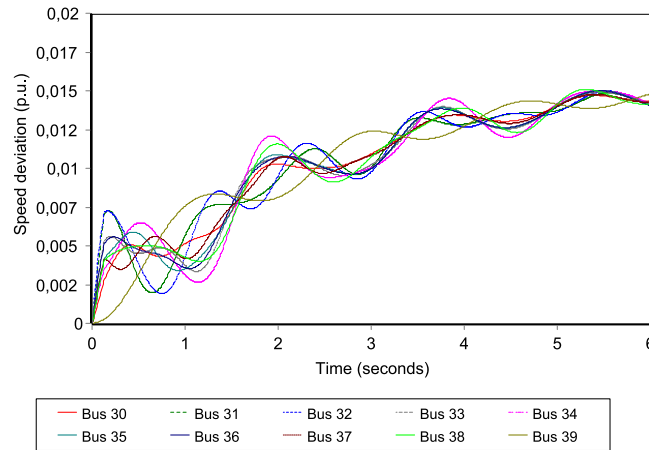


Fig. 8. Synchronous generator speed deviations.

Table 2

Co-simulator processing time (PSS/E time step $\Delta t = 0.01$ s).

	Longer than Δt		Shorter than Δt		
	←	→	←	→	
Time between events (s)	0.05	0.02	0.01	0.005	0.001
Total processing time (s)	0.70	1.73	3.03	3.88	3.94
Time (ms)/event	1.70	1.71	1.51	0.96	0.19

of communication events that roughly corresponds to the PSS/E time step, in this case $\Delta t = 0.01$ s, because it pauses PSS/E at every time step to process just one communication event.

To quantify this effect, Table 2 shows the processing times needed to solve a 10-seconds co-simulation depending on the time between communication events. To obtain this table, a series of dummy messages are sent between OMNeT++ modules to force the PSS/E simulation to pause.

Table 2 shows that up to one message each 0.01 s, which is the PSS/E time step, the computation time increases because the PSS/E simulation is forced to pause more often. From one message each 0.01 s onward the simulation time increases very slightly, while the ratio of simulation time per message decreases significantly.

Simulations have been run on a personal computer with a 2.5 GHz processor. The compilation of PSS/E and OMNeT++ prior to the execution of the simulation results in a solution that is faster than simulators based on interpreted languages. In all the cases the computation time is significantly shorter than the 86 s reported in [21] to simulate 0.5 s, while the results obtained are coherent.

5. Case study 2: Power system state estimation

5.1. Description

The aim of this case is to evaluate the effect of communication latency times on the performance of an online state estimator. The case represents a state estimator running in a control center and using data from phasor measurement units (PMUs) deployed throughout the IEEE 39 Bus Test System. For the purpose of the demonstration, the selected state estimation is based on an unscented Kalman filter (UKF), which is one of the mainstream algorithms used in power systems [39,40]. Measurements are sent to the control center through a communication channel and reach the state estimator after a certain latency time. The latency time of the communication channel affects the accuracy of the state estimator because if a measurement is delayed then the last available measurement is used as a best approximation, and it may not be consistent with the rest of the data.

The simulation represents a 3-phase short-circuit at bus 3, cleared after 150 ms. Fig. 9 shows the single-line diagram of the system together with the location of the fault and the measurements. Measurements include the magnitude of the voltage at several buses, the phase of the voltage at certain buses, and the active and reactive power produced by power plants, consumed at certain loads, and flowing through certain branches.

The measurements are modeled with a normally distributed random error. The standard deviation of the power and voltage magnitude and voltage angle measurements is, 3% and 0.7% of the reading value and $0.7 \cdot 10^{-2}$ rad, respectively.

The power system OMNeT++ module runs the simulation in PSS/E and sends the measurements via a communication channel to the state estimator, that receives each measurement after the corresponding latency time. Every measurement contains a time stamp indicating the time of the measurement, and the state estimator uses always the most recent measurements regardless of the time of their arrival.

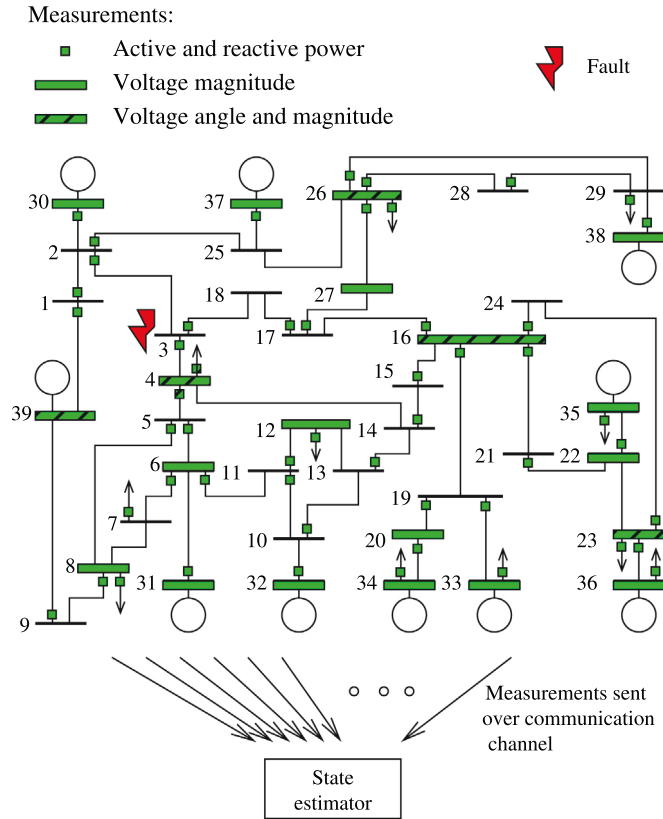


Fig. 9. Case study 2: Dynamic state estimation.

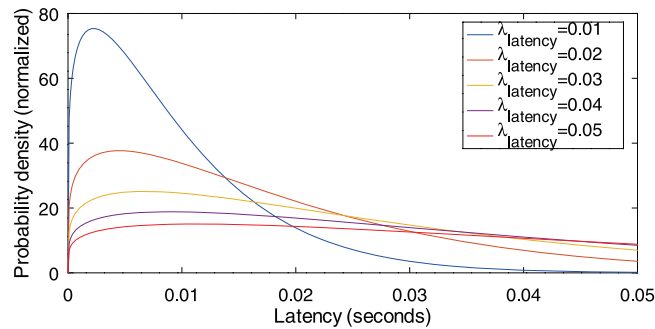


Fig. 10. Probability density functions used to model latency.

Each measurement is sent every 10 ms. There are in the literature different stochastic models of latency in wide-area power system signals [41,42]; in this study the communication latency is modeled as a Weibull probability distribution with shape parameter $k = 1.2$ and scale parameters $\lambda_{latency}$ ranging from 0.01 to 0.05, as shown in Fig. 10. The equation that defines the probability density function of the latency as a Weibull random variable is

$$f(x; \lambda_{latency}, k) = \frac{k}{\lambda_{latency}} \left(\frac{x}{\lambda_{latency}} \right)^{k-1} e^{-(x/\lambda_{latency})^k}. \tag{6}$$

The integration time step in PSS/E is set to 0.01 s.

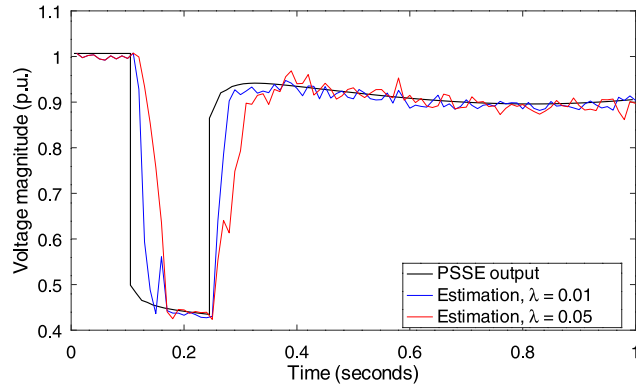


Fig. 11. Estimation of voltage magnitude at bus 17.

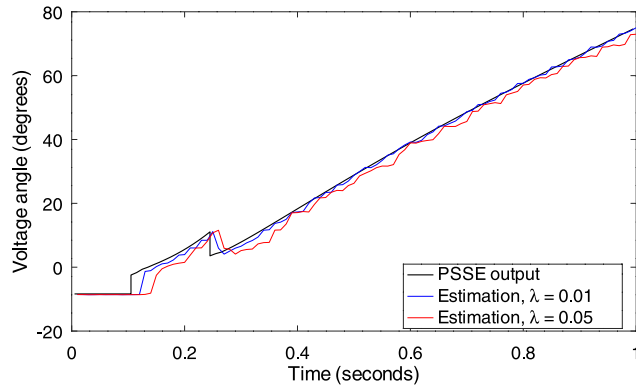


Fig. 12. Estimation of voltage angle at bus 17.

Table 3
State estimator residuals.

λ	Ψ_{δ} (rad)	Ψ_V (p.u.)
0.01	$2.870 \cdot 10^{-2}$	$5.602 \cdot 10^{-2}$
0.02	$3.340 \cdot 10^{-2}$	$6.151 \cdot 10^{-2}$
0.03	$3.942 \cdot 10^{-2}$	$7.326 \cdot 10^{-2}$
0.04	$4.599 \cdot 10^{-2}$	$7.922 \cdot 10^{-2}$
0.05	$5.073 \cdot 10^{-2}$	$8.368 \cdot 10^{-2}$

5.2. Results

The PSS/E simulation is compared with the output of the online state estimator to evaluate how the latency in the communications affect the estimation. As an example, Figs. 11 and 12 show the voltage and the angle in bus 17. It can be seen that the latency time in the communications cause a delay in the estimation with respect to the real value. The delay is more evident immediately after the set and the clearance of the fault, because these are moments when the measured variables change suddenly.

A second effect, less evident in the figures, is a decrease in the accuracy of the estimation when the latency time increases. This effect can be quantified by calculating the root mean square (RMS) of the residuals of the estimated voltage magnitudes (Ψ_V) and voltage angles (Ψ_{δ}). Table 3 shows how these values increase with the latency.

Using the estimated complex voltages it is possible to estimate other variables in the power system, for example the power produced by synchronous generators. Fig. 13 shows the active power produced by the generator connected to bus 30 as simulated in PSS/E and as calculated as a function of the results of the state estimator. Active power is calculated as

$$P_{30} = \text{Real} \left\{ V_{30} \left(\frac{V_{30} - V_2}{Z_{2,30}} \right)^* \right\}, \tag{7}$$

where V_{30} and V_2 are the estimated voltage phasors at buses 30 and 2 and $Z_{2,30}$ is the series impedance connecting buses 30 and 2. Fig. 13 shows that the state estimator provides a fair estimation of the active power produced by the power plant. Here again the latency time causes a delay and a decrease in the accuracy of the estimated values.

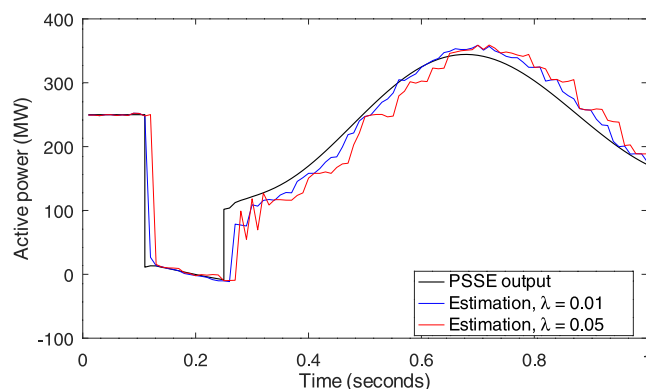


Fig. 13. Estimation of active power produced by the power plant at bus 30.

6. Conclusion

A co-simulator of power and communication systems based on OMNeT++ and PSS/E has been developed and demonstrated. The co-simulation platform makes it possible to integrate the simulation of communication systems with the extensive PSS/E library of component models and with the power system models already developed in PSS/E by TSOs.

From the point of view of the accuracy of the results, the proposed co-simulator applies an event-driven synchronization method that minimizes the errors caused by delays in the synchronization between both simulators.

From the point of view of the speed of the simulation, the total processing time is affected by the time separating communication events. The least efficient case corresponds to a coincidence between the time separating communication events and the power system numerical integration time step. However, even in this case the application of the method to the IEEE 39 Bus Test case shows a good performance. The fact that the co-simulation platform is compiled prior to the simulation, as opposed to simulators based on interpreted languages, results in a high computational speed.

Regarding the scalability of the co-simulator, the IEEE 39 Bus Test system is a relatively small case and the application to larger systems remains to be studied. However, given that PSS/E is specifically designed to simulate very large power systems, and that OMNeT++ facilitates performing large experiments, all signs indicate that the co-simulator will be suitable for large systems.

CRedit authorship contribution statement

Pablo Ledesma: Conceptualization, Methodology, Software, Writing – original draft. **Davide Gotti:** Methodology, Software, Validation, Writing – review & editing. **Hortensia Amaris:** Methodology, Writing – review & editing.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.compeleceng.2022.108092>.

Acknowledgments

This work was supported by the Spanish Agencia Estatal de Investigación under grant PID2019-104449RB-I00/AEI/10.13039/501100011033.

References

- [1] Li W, Zhang X. Simulation of the smart grid communications: Challenges, techniques, and future trends. *Comput Electr Eng* 2014;40(1):270–88.
- [2] Muller SC, Georg H, Nutaro JJ, Widl E, Deng Y, Palensky P, et al. Interfacing power system and ICT simulators: Challenges, state-of-the-art, and case studies. *IEEE Trans Smart Grid* 2018;9(1):14–24, [Online]. Available: <http://ieeexplore.ieee.org/document/7444194/>.
- [3] Palensky P, Van Der Meer A, Lopez C, Joseph A, Pan K. Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems. *IEEE Trans Ind Electron Mag* 2017;11(2):6–21, [Online]. Available: <http://ieeexplore.ieee.org/document/7956310/>.
- [4] Troiano GO, Ferreira HS, Trindade FCL, Ochoa LF. Co-simulator of power and communication networks using OpenDSS and OMNeT++. In: *IEEE PES innovative smart grid technologies conference Europe*. IEEE; 2016, p. 1094–9, [Online]. Available: <http://ieeexplore.ieee.org/document/7796538/>.
- [5] Awad A, Bazan P, German R. SGsim: A simulation framework for smart grid applications. In: *2014 IEEE international energy conference*. IEEE; 2014, p. 730–6, [Online]. Available: <http://ieeexplore.ieee.org/document/6850507/>, <http://www.scopus.com/inward/record.url?eid=2-s2.0-84904985762&partnerID=tzOtx3y1>.
- [6] Bhor D, Angappan K, Sivalingam KM. Network and power-grid co-simulation framework for smart grid wide-area monitoring networks. *J Netw Comput Appl* 2016;59:274–84, [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804515001447>.

- [7] Dede J, Kuladinithi K, Förster A, Nannen O, Lehnhoff S. OMNeT++ and mosaik: Enabling simulation of smart grid communications. In: Proceedings of the "OMNeT++community summit 2015". 2015, [Online]. Available: [arXiv:1509.03067](https://arxiv.org/abs/1509.03067).
- [8] Garau M, Celli G, Ghiani E, Soma GG, Pilo F, Corti S. ICT reliability modelling in co-simulation of smart distribution networks. In: 2015 IEEE 1st international forum on research and technologies for society and industry leveraging a better tomorrow. IEEE; 2015, p. 365–70, [Online]. Available: <http://ieeexplore.ieee.org/document/7325125/>.
- [9] Hammad E, Ezeme M, Farraj A. Implementation and development of an offline co-simulation testbed for studies of power systems cyber security and control verification. *Int J Electr Power Energy Syst* 2019;104:817–26, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0142061518309128>.
- [10] Shum C, Lau W-H, Mao T, Chung HS-H, Tsang K-F, Tse NC-F, et al. Co-Simulation of distributed smart grid software using direct-execution simulation. *IEEE Access* 2018;6:20531–44, [Online]. Available: <https://ieeexplore.ieee.org/document/8334243/>.
- [11] Venkatesh T, Jain T. Synchronized measurements-based wide-area static security assessment and classification of power systems using case based reasoning classifiers. *Comput Electr Eng* 2018;68:513–25.
- [12] GridPACK, [Online]. Available: <https://www.gridpack.org>.
- [13] Pan K, Teixeira A, Lopez CD, Palensky P. Co-Simulation for cyber security analysis: Data attacks against energy management system. In: 2017 IEEE international conference on smart grid communications. IEEE; 2017, p. 253–8, [Online]. Available: <http://ieeexplore.ieee.org/document/8340668/>.
- [14] Findrik M, Smith P, Kazmi JH, Faschang M, Kupzog F. Towards secure and resilient networked power distribution grids: Process and tool adoption. In: 2016 IEEE international conference on smart grid communications. IEEE; 2016, p. 435–40, [Online]. Available: <http://ieeexplore.ieee.org/document/7778800/>.
- [15] Ahmad I, Kazmi JH, Shahzad M, Palensky P, Gawlik W. Co-Simulation framework based on power system, AI and communication tools for evaluating smart grid applications. In: 2015 IEEE innovative smart grid technologies - Asia. IEEE; 2015, p. 1–6, [Online]. Available: <http://ieeexplore.ieee.org/document/7387092/>.
- [16] Svenda VG, Stankovic AM, Saric AT, Transtrum MK. Influence of communication irregularities and Co-simulation on hybrid power system state estimation. In: Proceedings - 2018 IEEE PES innovative smart grid technologies conference Europe. Institute of Electrical and Electronics Engineers Inc.; 2018.
- [17] Milano F. Power system modelling and scripting, vol. 54. Springer-Verlag; 2010, p. 1–550.
- [18] Borodulin MY. Effect of numerical integration on critical time evaluation in power system stability studies. In: IEEE power and energy society general meeting. 2013.
- [19] Ramakrishna DS. Numerical analysis. Pearson; 2014, p. 768, [Online]. Available: <https://learning.oreilly.com/library/view/numerical-analysis-1-e/9789332540804/>.
- [20] Zeigler BP, Muzy A, Kofman E. Theory of modeling and simulation : Discrete event and iterative system computational foundations. Academic Press; 2018, p. 667.
- [21] Lin H, Veda SS, Shukla SS, Mili L, Thorp J. GECO: GLObal event-driven co-simulation framework for interconnected power system and communication network. *IEEE Trans Smart Grid* 2012;3(3):1444–56, [Online]. Available: <http://ieeexplore.ieee.org/document/6200399/>.
- [22] OMNeT++ discrete event simulator, [Online]. Available: <https://omnetpp.org/>.
- [23] Li W, Ferdowsi M, Stevic M, Monti A, Ponci F. Cosimulation for smart grid communications. *IEEE Trans Ind Inf* 2014;10(4):2374–84.
- [24] Kundur P. Power system stability and control. McGraw-Hill; 1994, p. 1176.
- [25] Ledesma P. Example in co-simulation platform for interconnected power systems and wide area communication networks. 2021, <http://dx.doi.org/10.21950/HCWVYJ>.
- [26] Le TD, Anwar A, Beuran R, Loke SW. Smart grid co-simulation tools: Review and cybersecurity case study. In: 7th international conference on smart grid. Institute of Electrical and Electronics Engineers Inc.; 2019, p. 39–45.
- [27] Li X, Huang Q, Wu D. Distributed large-scale co-simulation for IoT-aided smart grid control. *IEEE Access* 2017;5:19951–60.
- [28] Hopkinson K, Wang X, Giovanini R, Thorp J, Birman K, Coury D. EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. *IEEE Trans Power Syst* 2006;21(2):548–58.
- [29] Li W, Monti A, Luo M, Dougal RA. VNET: A co-simulation framework for analyzing communication channel effects on power systems. In: 2011 IEEE electric ship technologies symposium. 2011, p. 143–9.
- [30] Liberatore V, Al-Hammouri A. Smart grid communication and co-simulation. In: IEEE 2011 EnergyTech. 2011.
- [31] Nutaro J. Designing power system simulators for the smart grid: Combining controls, communications, and electro-mechanical dynamics. In: IEEE power and energy society general meeting. 2011.
- [32] Lévesque M, Xu DQ, Maier M, Joás G. Communications and power distribution network co-simulation for multidisciplinary smart grid experimentations communications and power distribution network co-simulation for multidisciplinary smart grid experimentations *. In: Proc. 45th annu. simulation symp. 2012. p. 2:1–2:7.
- [33] Georg H, Müller SC, Rehtanz C, Wietfeld C. Analyzing cyber-physical energy systems: The INSPIRE cosimulation of power and ICT systems using HLA. *IEEE Trans Ind Inf* 2014;10(4):2364–73.
- [34] Celli G, Pegoraro PA, Pilo F, Pisano G, Sulis S. DMS Cyber-physical simulation for assessing the impact of state estimation and communication media in smart grid operation. *IEEE Trans Power Syst* 2014;29(5):2436–46.
- [35] Wei M, Wang W. Greenbench: A benchmark for observing power grid vulnerability under data-centric threats. In: Proceedings - IEEE INFOCOM. Institute of Electrical and Electronics Engineers Inc.; 2014, p. 2625–33.
- [36] Garau M, Celli G, Ghiani E, Pilo F, Corti S. Evaluation of smart grid communication technologies with a Co-simulation platform. *IEEE Wirel Commun* 2017;24(2):42–9.
- [37] Mana PT, Schneider KP, Du W, Mukherjee M, Hardy T, Tuffner FK. Study of microgrid resilience through Co-simulation of power system dynamics and communication systems. *IEEE Trans Ind Inf* 2021;17(3):1905–15.
- [38] Deng X, Jiang Z, Sundaresh L, Yao W, Yu W, Wang W, et al. A time-domain electromechanical co-simulation framework for power system transient analysis with retainment of user defined models. *Int J Electr Power Energy Syst* 2021;125:106506.
- [39] Liu H, Hu F, Su J, Wei X, Qin R. Comparisons on Kalman-filter-based dynamic state estimation algorithms of power systems. *IEEE Access* 2020;8:51035–43.
- [40] Gotti D, Ledesma P, Amaris H. Comparative analysis between state estimation algorithms under static and dynamic scenarios. In: Proceedings - 2020 IEEE international conference on environment and electrical engineering and 2020 IEEE industrial and commercial power systems Europe, EEEIC / I and CPS Europe 2020. Institute of Electrical and Electronics Engineers Inc.; 2020.
- [41] Zhu K, Chenine M, Nordström L, Holmström S, Ericsson G. Design requirements of wide-area damping systems - using empirical data from a utility IP network. *IEEE Trans Smart Grid* 2014;5(2):829–38.
- [42] Stahlhut JW, Browne TJ, Heydt GT, Vittal V. Latency viewed as a stochastic process and its impact on wide area power system control signals. *IEEE Trans Power Syst* 2008;23(1):84–91.

Pablo Ledesma received his Ph.D. in 2001 at University Carlos III of Madrid, where he is currently an Associate Professor. He has worked with the Spanish TSO on several projects on large-scale integration of renewable energy. His areas of research are transient stability and dynamic modeling of power systems.

Davide Gotti (S'21) received the B.Sc. and M.Sc. degrees in electrical engineering from Politecnico di Milano, Italy, and the M.Sc. degree in renewable energy in electrical systems from Universidad Carlos III de Madrid in 2018. His research interests include state estimation, power system modeling and machine learning algorithms.

Hortensia Amaris (M00-SM19) received the Ph.D. degree from the Technical University of Madrid (UPM), Madrid, in 1995. She is currently a Full Professor of electrical engineering with University Carlos III of Madrid, Spain, where she has been working mainly on operation and control of power networks, power quality, power-electronic converters, renewable energy sources and smart grids.