Using Co-Simulation for the Integrated Planning and Analysis of Wide Area Measurement Systems

Halil Alper Tokel, Gholamreza Alirezaei, Thomas Salzmann, and Rudolf Mathar
Institute for Theoretical Information Technology
RWTH Aachen University, D-52056, Aachen, Germany
{tokel,alirezaei,salzmann,mathar}@ti.rwth-aachen.de

***Abstract*—The increasing penetration of measurement and communication infrastructures in power grids has led to the development of new applications and use cases in the utility sector, such as demand side management and virtual power plants. The success of such applications, however, depends on a proper interplay of the underlying communication network and the power grid. For this reason, integrated planning approaches for the extension of measurement and communication infrastructures in the power grids have recently been a hot research topic. Although such planning techniques are of crucial significance for an optimum network design satisfying the requirements of both the power system and the communication network, a simulative performance analysis must accompany the planning process. Therefore, in this work we present a co-simulation environment and tool chain to enable integrated planning and subsequent performance analysis of a wide area measurement system. As an example use case, the proposed environment is used to investigate the accuracy of phasor measurement unit-based linear state estimation techniques for IEEE 14-bus test network under several communication network scenarios.**

I. Introduction

Recent advances in information and communication technologies (ICT) have stimulated significant developments in the utilities sector thanks to advanced monitoring and data analysis techniques. As a result of the increasing penetration of measurement and communication infrastructures, which are called wide area measurement systems (WAMS), into distribution grids, new applications and concepts have arisen, such as demand side management or virtual power plants [1]. On the one hand, these developments pave the way for the successful integration of renewables and a more efficient grid operation. However, on the other hand, the planning of grid expansion and the operation of the grid have become more complex tasks as a result of the interdependencies between the power grid, the ICT infrastructure, and new business models. The concerns on reliability and security of the future grid necessitate, therefore, interdisciplinary approaches for the development of new technologies, algorithms, and applications in all three domains of communication networks, power systems, and market design.

In our recent work, we proposed a novel optimization model which enables a minimum-cost design of a WAMS with a hierarchical heterogenous communication network [2]. The proposed approach delivers both the required number and locations of phasor data concentrators (PDC) and phasor measurement units (PMU) for observability of the whole power system, and a hierarchical heterogeneous communication network design under data communication requirements in alignment with *IEEE Standard for Synchrophasor Data Transfer for Power Systems* [3]. Although planning techniques, such as the one presented in [2], are valuable as a structured approach, it is of crucial importance to further analyze the system performance in numerical simulations considering different network scenarios. In this context, the integration of multiple individual domain-specific and powerful simulators as *a cooperative simulation environment (co-simulation)* for power system simulations has been recently proposed by numerous researchers. The advantage of a co-simulation is the possibility to use thorough and precise models which are available in individual simulators in order to create an overall close-to-reality mathematical model of the system components. One of the early works in this context is [4], which proposed a run-time environment bridging independent power system and communication network simulators. [5] brings forward the idea of *global event-driven co-simulation* for smart grid simulations. Recently, the increasing importance of WAMS for the success of future applications led the WAMS simulation to be one of the key fields where similar co-simulation approaches proved valuable. For instance, [6] uses the global event list approach of [5] to analyze the effects of communication network on the performance of wide area protection. Similarly, [7] presents a test-bed for the investigation of communication link delays on the wide area monitoring, whereas [8] uses a co-simulation environment for the analysis of state estimation in a WAMS with a WiMAX communication network. The mentioned studies conclude in consensus that the interdependencies between the power grid applications and the communication network must be considered in the planning phase.

Therefore, the main aim of this paper is to present a co-simulation environment and tool chain which enables an integrated planning and a subsequent simulative performance analysis of the designed WAMS. Furthermore, as a use case, we investigate the effect of several communication network scenarios on the accuracy of the PMU-based linear state estimation techniques in the WAMS with the introduced co-simulation environment.

The rest of this paper is structured as follows: We start with some background information about the WAMS and its network architecture. Next, we introduce the entire co-simulation environment used in this work, consisting of OMNeT++ [9] as the communication network simulator and MatDyn [10] as the dynamic power system simulator along with the co-simulation interface MOSAIK [11]. In this context, we intend to provide researchers from various fields with a flexible solution to extend OMNeT++ for similar co-simulation requirements. In Section V, we use the proposed tool chain for the planning and co-simulation of a WAMS, whose topology is obtained with the optimization model in [2] for the IEEE 14-bus network. Particularly, we investigate the effect of the communication link delays and node failures on the accuracy of the power system state estimation, as well as the capability of the state estimators to the changes in the power system state. Finally, Section VI concludes this paper with a summary and some remarks on our future work.

II. WAMS Architecture and Planning

A WAMS consists of *i)* measurement devices, called phasor measurement units (PMU), which measure the voltage and current phasor values available at the system nodes where they are installed, *ii)* several data concentrator units, called phasor data concentrators (PDC), and *iii)* a data processing center, called SuperPDC (SPDC). *IEEE Standard for Synchrophasor Data Transfer for Power Systems* [3] lays down the architecture for the communication network in a WAMS as shown in Figure 1. This architecture postulates a hierarchical transmission of sensor data from PMUs to PDCs, where a preprocessing of the data takes place such as time alignment and consistency check [2]. PDCs send the data to a central unit SPDC, where the measurement data from a larger part of the network are aggregated to execute energy management functions such as state estimation, cf. [12].

The planning approach presented in [2] reveals i) the required number and exact locations of PMU and PDCs for full observability of the power system, ii) the required communication network, which includes the locations and capabilities of necessary telecommunication equipment to install along with required links and their technologies, iii) a guarantee for the fulfillment of the capacity and delay specifications, and iv) insights about the operation of the network, such as the utilization of the communication links and the overall robustness and the reliability of the network. For the details of the mathematical optimization model, please refer to [2]. In the next section, we introduce a co-simulation framework which enables the integrated planning of a WAMS and its subsequent simulative performance analysis.

III. Simulation Tools and co-simulation Environment

We start with an introduction of OMNeT++ and MatDyn, and then present the co-simulation environment with the co-simulation interface MOSAIK.

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators [9]. It is a discrete-event simulation (DES) tool, in which the simulation time progresses as the discrete simulation events, kept in an event list, are executed in the order of execution. An event can be any interaction or change in the system, such as an arrival of a packet or an expiry of a timer. The execution of events are governed by an *event scheduler*. The default scheduler in OMNeT++ is the sequential scheduler which executes all the events sequentially until the end of the simulation.

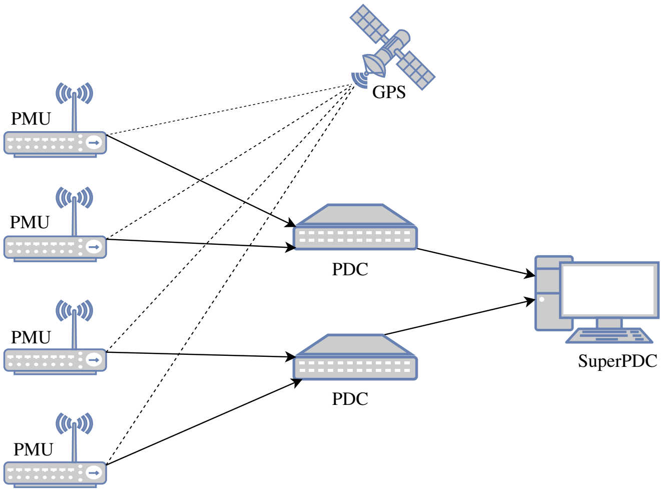

Fig. 1. Hierarchical network architecture of WAMS. PMUs send the phasor measurements, time-stamped by the GPS signal, to a SuperPDC over intermediate PDCs [3].

MatDyn is a free Matlab based open source program to perform dynamic analysis of electric power systems [10]. It is based on the power flow and optimal power flow solutions by the steady state power system analysis toolbox MATPOWER [13]. The simulation in MatDyn progresses as the system of differential algebraic equations (DAE), which governs the power system, is solved iteratively to calculate the system parameters at the next step depending on the generator and load models. For the details of the mathematical models used for the generators and the solutions of DAE systems, please refer to [10].

The most critical issue in a co-simulation with a dynamic power system simulator like MatDyn and a communication network simulator like OMNeT++ is the implementation of a time synchronization mechanism which would ensure a correct progress of the simulation time and a timely data exchange between the simulators. The underlying reason is the different natures of both simulators regarding the progress of the simulation time. The simulation time in a dynamic power system simulation advances with fixed intervals due to the iterative solution of a system of differential algebraic equations. In contrast, the simulation time in a discrete-event communication network simulator progresses with the execution of events which might be unevenly distributed on the time axis. Figure 2 depicts the flow of simulation time in both simulators. Several researchers have recently dealt with the selection of a proper synchronization approach for smart grid simulations. [14] provides an overview of several approaches and a comparison between them in terms of run time and

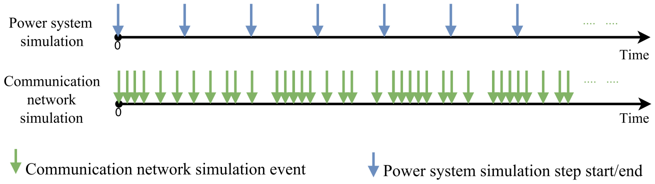
Power system simulation
0
Time
Communication network simulation
0
Time
Communication network simulation event
Power system simulation step start/end

Fig. 2. The advance of simulation time in dynamic power system simulator MatDyn and discrete-even communication network simulator OMNeT++

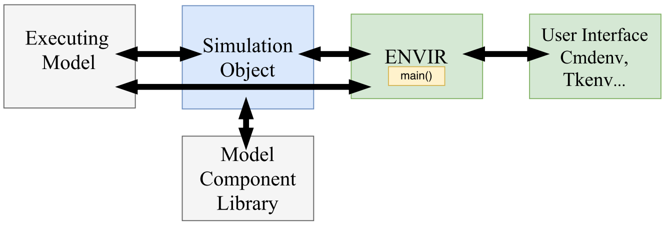

Fig. 3. Modular software architecture of OMNeT++, adapted from [16]

scalability. In [5], a global event list has been proposed for the consistent execution of events by the co-simulation interface. This approach, however, requires the implementation of a global event management interface. On the other hand, the most intuitive way to realize a synchronization mechanism is to run both simulators for a certain step size independently and then facilitate the required data exchange between the simulators at the end of each step as adopted by the tools EPOCHS [4] and MOSAIK [11]. This approach not only reduces the programming effort to create a run-time infrastructure for the event-management but also enables a trade-off between the simulation run time and the accuracy of the co-simulation with the selection of a proper step size. In the current paper, we use the tool MOSAIK due to its convenience and the available detailed documentation [15].

MOSAIK provides an API to enable a communication and data exchange with and between simulators through TCP sockets and messages encoded in JSON data format. Therefore, it is necessary to extend the available simulators with this interface and message handling functionalities in order to use them in a co-simulation with MOSAIK. Whereas MatDyn can be extended relatively easily for a co-simulation with MOSAIK due to its transparent structure, the extension of OMNeT++ requires a deeper understanding of its software architecture. Therefore, we briefly discuss the extension of OMNeT++ in the following section.

IV. Extension of OMNeT++

The modular architecture of OMNeT++ is depicted in Figure 3 along with the different libraries of OMNeT++ simulation environment and their relations. In Figure 3, the *model component library* contains definitions of modules, channels, networks, classes along with their implementation which can be used in a simulation, and *executing model* represents the class instances which are created in the particular simulation. *Simulation object* is the simulation kernel and the class library, whereas *envir* is the base environment library which is common to all implemented user interfaces, such as *command environment (cmdenv)* and *graphical user environment (tkenv)* shown in the rightmost box.

In order to enable a flexible solution for the co-simulation with MOSAIK, we extend the main() function of OMNeT++ and introduce a new base environment class along with a new user interface class. The proposed extended architecture is shown in Figure 4. The new simulation environment class *cosimulation envir* and the cosimulation user interface *cosimenv* enable the progress of the simulation in steps of a step size determined by the co-simulation interface MOSAIK in a similar way which was proposed in [17]. A step is implemented by a *simulateStep()* function in *cosimenv*, which executes all scheduled simulation events until the next synchronization point. After the execution of each simulation step, a data exchange for the co-simulation synchronization takes place via the TCP socket interface.

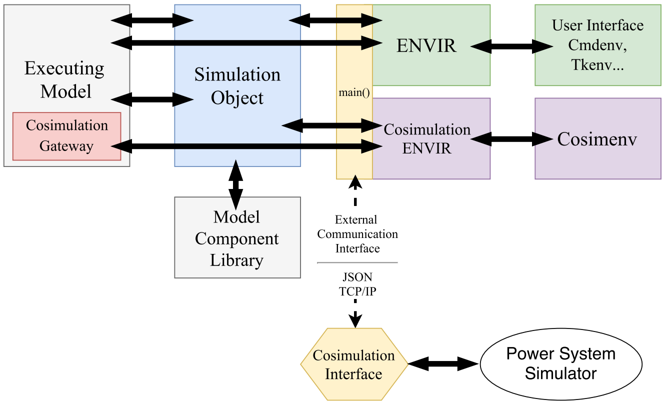

Fig. 4. Extended modular software architecture of OMNeT++. A co-simulation environment and a co-simulation user interface class have been implemented, along with the external communication interface over TCP sockets, extended based on [16].

The external input received from MOSAIK, which contains, for example, the next synchronization time and the new attribute values of module parameters, such as measurement values from the power system, is received in JSON format on the TCP socket. Any external input is here imaginable depending on the investigated scenario. For example, communication link and node failures, or changes in the system parameters can be introduced to investigate their impacts on the network. This external input is then passed by the *co-simulation envir* class to a network module, which we call *co-simulation gateway* and is located in *the executing model* referring to Figure 4, by updating one of its parameters with the JSON string value of the external input received at the synchronization time. This solution was chosen to benefit from the *handleParameterChange()* method of *cComponent* class, from which all other OMNeT++ simulation modules inherit. *handleParameterChange()* method of a network module is called when a parameter of the module is changed by another mechanism, in this case by the introduced *co-simulation user interface* class. Thus, this method of the *co-simulation gateway* class is implemented in such a way that it reads the external input from its parameter and notifies all other necessary modules in the network through a direct message including relevant information right before the start of the next step. This relatively simple approach enables a structured communication between the co-simulation interface and any network module in the *executing model* referring to Figure 4.

As for the messages from OMNeT++ towards the co-simulation interface, the same procedure takes place in the reverse order. The messages from individual modules are passed to the *co-simulation gateway* via direct messages. After each step, the simulation environment checks for a possible output message in the *co-simulation gateway* and sends it to the co-simulation interface. As a result of this setting, the requirements for a co-simulation with MOSAIK are effectively and efficiently fulfilled. Moreover, all inherent functionalities of OMNeT++ can be used in the new co-simulation environment.

V. Design and Simulation of a WAMS for IEEE 14-Bus Test Network

In this section, we describe the steps in the planning and simulation of a WAMS for the IEEE 14-bus test network with the presented co-simulation environment [18]. The network data are provided as an input in the MOSAIK script, and both simulators are initialized with the given network topology in their domains. The planning step of WAMS, i.e., the solution of the mathematical model in [2], takes place in OMNeT++ with all relevant constraints regarding the telecommunication technologies and the system requirements, using Gurobi 7.0 as the solver [19]. After this step, the PMU locations are conveyed to MatDyn through the co-simulation interface MOSAIK.

MOSAIK governs the flow of the co-simulation with a step-wise execution of individual simulators, where at each synchronization point, the power system simulator sends the measurement values to the communication network simulator. The measurement values are then sent to the SPDC node in OMNeT++ where the state estimation takes place. Any control command from this module can be then sent back to the other communication network nodes within OMNeT++.

A. Use Case : PMU-Based Linear State Estimation

As a concrete example use case, we investigate the accuracy of two PMU-based linear state estimation techniques using the introduced co-simulation environment. Therefore, in the following, we briefly introduce the two techniques, namely linear weighted least square (LWLS) state estimation and discrete Kalman filter (DKF) state estimation for one-phase, based on the assumption of a balanced 3-phase system [20].

We define the system state $\boldsymbol{x}$ of a power grid with n buses by the vector $\boldsymbol{x} = [V_{1,re}, \dots, V_{n,re}, V_{1,im}, \dots, V_{n,im}]$, where $V_{n,re}$ and $V_{n,im}$ are the real and the imaginary parts of the voltage phasor V_n at node n, respectively. The vector of measurements is denoted by

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{H}$ is the measurement matrix, which describes the linear relation between the measurement i and the state vector $\boldsymbol{x}$, while $\boldsymbol{e}$ is the vector of measurement error. It is assumed that the measurement errors are independent and zero-mean Gaussian distributed, i.e., $\boldsymbol{e} \sim \mathcal{N}(0, \boldsymbol{R})$, where $\boldsymbol{R} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_n^2)$ is the covariance matrix, with the variances σ_i^2 of the noise components e_i as its diagonal entries.

1) Linear Weighted Least Square State Estimation: The LWLS estimator tries to find the state vector, which minimizes the weighted sum of the squared error in the measurements as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad J(\boldsymbol{x}), \tag{2}$$

where $J(\boldsymbol{x})$ is defined by $(\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x})^T \boldsymbol{R}^{-1} (\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x})$. The analytical solution of (2) is calculated as

$$\hat{\boldsymbol{x}}_{\text{LWLS},k} = \boldsymbol{G}^{-1} \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{z}_k, \tag{3}$$

where index k is the time index, and $\boldsymbol{G} = \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}$. Note that in this particular work $\boldsymbol{R}$ and therefore $\boldsymbol{G}$ do not vary over time.

2) Discrete Kalman Filter State Estimation: The general DKF is a widely used filter to estimate the state of a system which can be described by the equations

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \tag{4a}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{4b}$$

where $\boldsymbol{A}$ is the matrix that relates $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k-1}$, $\boldsymbol{B}$ is the matrix that relates the control input $\boldsymbol{u}$ to the next system state $\boldsymbol{x}_k$, and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the measurement and process noises with known covariance matrices $\boldsymbol{R}$ and $\boldsymbol{Q}_k$. Note that $\boldsymbol{v}$ and $\boldsymbol{w}$ are assumed to have zero cross-correlation.

DKF makes use of *a priori* estimate $\tilde{\boldsymbol{x}}_k$ that can be calculated based on the last estimate $\hat{\boldsymbol{x}}_{k-1}$ and the available system information. Thus, the prediction error $\tilde{\boldsymbol{e}}_k$ and the estimation error $\boldsymbol{e}_k$ are defined as

$$\tilde{\boldsymbol{e}}_k = \boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k, \tag{5a}$$

$$\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k, \tag{5b}$$

with corresponding prediction error covariance matrix $\tilde{\boldsymbol{P}}_k = \mathcal{E}(\tilde{\boldsymbol{e}}_k \tilde{\boldsymbol{e}}_k^T)$ and estimation error covariance matrix $\boldsymbol{P}_k = \mathcal{E}(\boldsymbol{e}_k \boldsymbol{e}_k^T)$, where $\boldsymbol{x}_k$ denotes the true system state. The estimation of the next system state consists of two steps. In the first step, which is called *prediction*, a priori prediction $\tilde{\boldsymbol{x}}_k$ for the next system state and the error covariance matrix are calculated as

$$\tilde{\boldsymbol{x}}_k = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k, \tag{6a}$$

$$\tilde{\boldsymbol{P}}_k = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q}_{k-1}. \tag{6b}$$

The estimation step then follows as

$$\boldsymbol{K}_k = \tilde{\boldsymbol{P}}_k \boldsymbol{H}^T (\boldsymbol{H}\tilde{\boldsymbol{P}}_k \boldsymbol{H}^T + \boldsymbol{R})^{-1}, \tag{7a}$$

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \boldsymbol{K}_k (\boldsymbol{z}_k - \boldsymbol{H}\tilde{\boldsymbol{x}}_k), \tag{7b}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H})\tilde{\boldsymbol{P}}_k, \tag{7c}$$

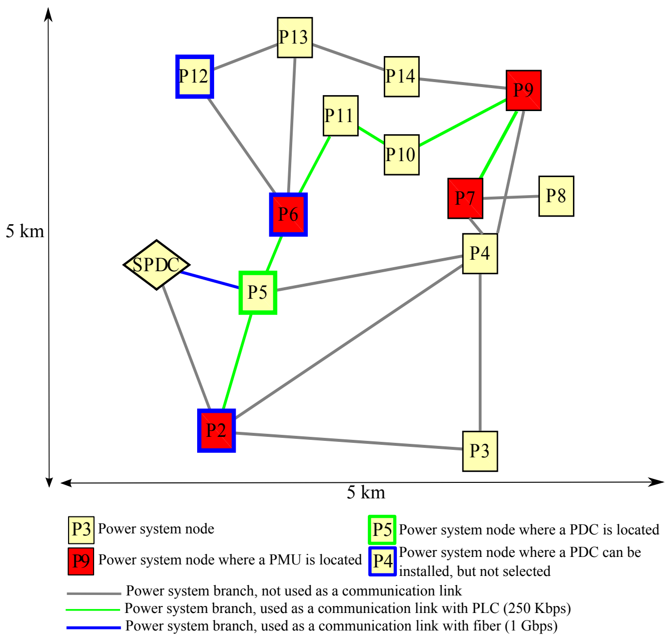
5 km
5 km
P13
P12
P14
P9
P11
P10
P7
P8
P6
SPDC
P4
P5
P2
P3
P3 Power system node
P9 Power system node where a PMU is located
P5 Power system node where a PDC is located
P4 Power system node where a PDC can be installed, but not selected
Power system branch, not used as a communication link
Power system branch, used as a communication link with PLC (250 Kbps)
Power system branch, used as a communication link with fiber (1 Gbps)

Fig. 5. The minimum cost WAMS topology with the optimum PMU. PDC locations and the communication links, obtained by the optimization model in [2]

where $\boldsymbol{K}$ is called Kalman gain. In this work, we adopt the auto regressive integrated moving average (ARIMA) process model, i.e., $\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{B} = \boldsymbol{0}$ and that the last state estimate is a good approximation for the next system state as discussed in [21] and [20].

B. WAMS Topology and Simulation Parameters

The minimum-cost WAMS topology for the IEEE 14-bus test network is shown in Figure 5. Note that the node locations are not included in the power system data [18], but approximately generated in this work on a region of $5\,\mathrm{km} \times 5\,\mathrm{km}$ for the design of a communication network. Furthermore, the planning is based on the assumption of available power line communication (PLC) links with a capacity of $250\,\mathrm{Kbps}$ and fiber links with a capacity of $1\,\mathrm{Gbps}$. For further cost assumptions, please refer to [2].

The traffic generation parameters of a PMU in the communication network are chosen in alignment with the *IEEE Standard for Synchrophasor Data Transfer for Power Systems* [3] with a measurement frequency of $50\,\mathrm{Hz}$ and the overhead of UDP/IP layers. The step size of the co-simulation and the power system simulation is set to $10\,\mathrm{ms}$, and a state estimation of the power system has been performed each ms at the SPDC. The PMU measurement errors are assumed to be independent with an SNR of $30\,\mathrm{dB}$.

The accuracy of the state estimation at time instance k is assessed with the metric of total vector error (TVE) defined in [3] as

$$TVE_k = \frac{|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_{\mathrm{true},k}|}{|\boldsymbol{x}_{\mathrm{true},k}|}, \tag{8}$$

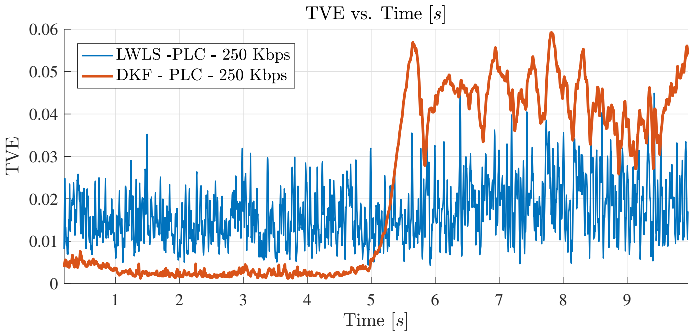

Fig. 6. A sudden load increase of 10% is applied at node P9 at t=5s. DKF performance degrades due to the abrupt change.

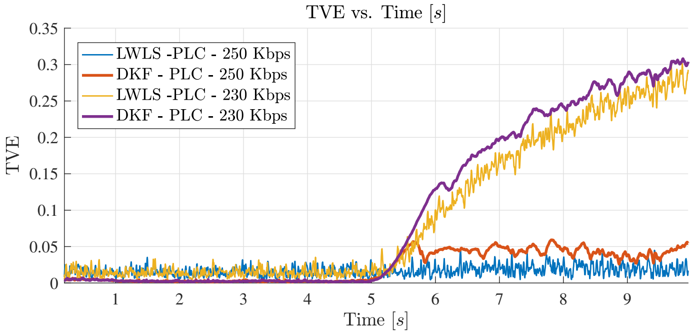

Fig. 7. A sudden load increase of 10% is applied at node P9 at t=5s. Lower data rate at the PLC links leads to a deterioration in the accuracy of state estimation.

where $\boldsymbol{x}_{\mathrm{true},k}$ the actual state vector at time instance k, and $\hat{\boldsymbol{x}}_k$ is the estimated state vector at the SPDC at time instance k.

The simulation duration is set to $10\,\mathrm{s}$ during which a sudden load increase of 10% is applied at node P9. This scenario is then simulated with the following communication network scenarios : *i)* PLC links with a data rate of $250\,\mathrm{Kbps}$, which is used in the optimum topology, *ii)* PLC links with $230\,\mathrm{Kbps}$, *iii)* a router failure at node P6 between t = $4\,\mathrm{s}$ and t = $6\,\mathrm{s}$, *iv)* bit error rate (BER) of $1\mathrm{x}10^{-6}$ and $2.5\mathrm{x}10^{-4}$ on PLC links with $250\,\mathrm{Kbps}$ data rate. In the next section, we present and discuss the accuracy of the state estimation observed in the simulations.

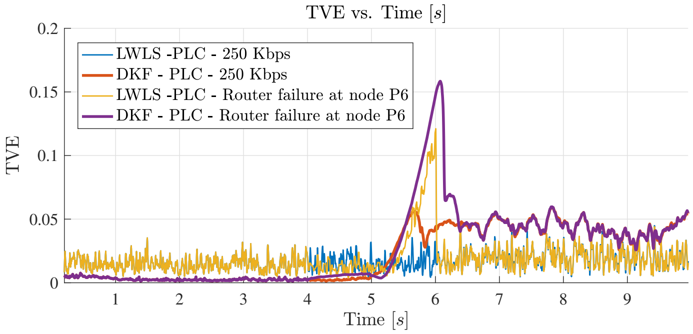

Fig. 8. A sudden load increase of 10% is applied at node P9 at t=5s, where the router at node P6 fails between t=4s and t=6s.

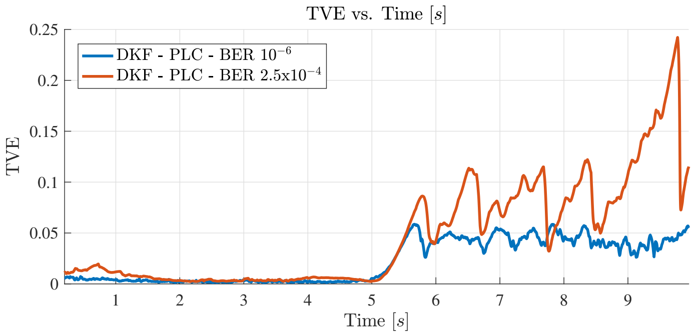

Fig. 9. A sudden load increase of 10% is applied at node P9 at t=5s. Increasing BER on PLC links leads to a degradation of state estimation accuracy.

C. Results & Discussion

Figure 6 shows the accuracy of LWLS and DKF state estimators where the PLC links are simulated with a data rate of 250 Kbps as in the optimum topology. We observe that DKF performs significantly superior until the abrupt load change at t = 5 s in alignment with the results in [20], however, its performance degrades significantly afterwards. Figure 7 illustrates the degradation in the state estimation accuracy of both techniques in case the PLC links had a data rate of 230 Kbps. The reason of this degradation is the congestion on the PLC link between nodes P6 and P5 which leads the measurement packets from P6, P7, and P9 to experience a significantly higher delay.

The impacts of a router failure at node P6 between t = 4 s and t = 6 s, and an increased BER on PLC links are shown in Figure 8 and Figure 9, respectively. We observe that the estimation accuracy is significantly influenced by the failures and performance degradation in the communication network.

Note that the state estimate at SPDC is the basis for the control actions to be applied in the power grid operation. The correct interpretation of the state estimate along with the information about the packet delays is of crucial significance for a proper system operation. Therefore, the state estimators and the wide area protection and control algorithms must take into account also the communication network performance.

VI. Conclusion

In this work, we have introduced a cosimulation environment and tool chain for the integrated planning and subsequent integrated simulative performance analysis of a WAMS. As an example application, the impact of communication network performance and failures on the state estimation accuracy has been investigated. The proposed tool provides a useful framework for future work in the development and analysis of distributed state estimation and fault detection algorithms under consideration of the interdependence between power grids and communication networks.

References

[1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: Communication technologies and standards,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov 2011.

[2] H. A. Tokel, G. Alirezaei, and R. Mathar, “Integrated network design for measurement and communication infrastructures in smart grids,” in *26th International Telecommunication Networks and Applications Conference (ITNAC 2016)*, Dunedin, New Zealand, Dec. 2016, pp. 267–273. [Online]. Available: http://www.ti.rwth-aachen.de/publications/output.php?id=1055\&table=proceeding\&type=pdf

[3] “IEEE Standard for Synchrophasor Data Transfer for Power Systems,” *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, Dec 2011.

[4] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, “EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components,” *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 548–558, 2006.

[5] H. Lin, S. S. Veda, S. S. Shukla, L. Mili, and J. Thorp, “GECO: Global event-driven co-simulation framework for interconnected power system and communication network,” *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1444–1456, Sept 2012.

[6] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, “A study of communication and power system infrastructure interdependence on pmu-based wide area monitoring and protection,” in *2012 IEEE Power and Energy Society General Meeting*, July 2012, pp. 1–7.

[7] K. Zhu, M. Chenine, and L. Nordstrom, “ICT architecture impact on wide area monitoring and control systems' reliability,” *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2801–2808, Oct 2011.

[8] G. Celli, P. A. Pegoraro, F. Pilo, G. Pisano, and S. Sulis, “DMS cyber-physical simulation for assessing the impact of state estimation and communication media in smart grid operation,” *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2436–2446, Sept 2014.

[9] A. Varga, “The OMNeT++ discrete event simulation system,” in *Proceedings of European Simulation Multiconference (ESM 2001)*, editor, Ed., June.

[10] S. Cole and R. Belmans, “MatDyn, a new Matlab-based toolbox for power system dynamic simulation,” *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1129–1136, Aug 2011.

[11] S. Schütte, S. Scherfke, and M. Tröschel, “Mosaik: A framework for modular simulation of active components in smart grids,” in *2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, Oct 2011, pp. 55–60.

[12] H. A. Tokel, G. Alirezaei, and R. Mathar, “Modern heuristical optimization techniques for power system state estimation,” in *The 2016 International Conference on Swarm Intelligence Based Optimization*, June 2016, pp. 78–85. [Online]. Available: https://www.ti.rwth-aachen.de/publications/output.php?id=1036&table=proceeding&type=pdf

[13] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[14] S. Ciraci, J. Daily, K. Agarwal, J. Fuller, L. Marinovici, and A. Fisher, “Synchronization algorithms for co-simulation of power grid and communication networks,” in *2014 IEEE 22nd International Symposium on Modelling, Analysis Simulation of Computer and Telecommunication Systems*, Sept 2014, pp. 355–364.

[15] mosaik 2.3.0 documentation. OFFIS Institut für Informatik. [Online]. Available: http://mosaik.readthedocs.io/en/latest/

[16] OMNeT++ simulation manual, OMNeT++ version 5.0. [Online]. Available: https://omnetpp.org/doc/omnetpp/manual/

[17] J. Dede, K. Kuladinithi, A. Förster, O. Nannen, and S. Lehnhoff, “OMNeT++ and mosaik: Enabling simulation of smart grid communications,” *arXiv preprint arXiv:1509.03067*, 2015. [Online]. Available: http://arxiv.org/abs/1509.03067

[18] “Power systems test case archive - UWEE.” [Online]. Available: https://www.ee.washington.edu/research/pstca/

[19] Gurobi Optimization Inc., “Gurobi optimizer reference manual,” 2015. [Online]. Available: http://www.gurobi.com

[20] S. Sarri, L. Zanni, M. Popovic, J. Y. L. Boudec, and M. Paolone, “Performance assessment of linear state estimators using synchrophasor measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 535–548, March 2016.

[21] L. Zanni, S. Sarri, M. Pignati, R. Cherkaoui, and M. Paolone, “Probabilistic assessment of the process-noise covariance matrix of discrete kalman filter state estimation of active distribution networks,” in *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, July 2014, pp. 1–6.