A Co-Simulation Framework for Integrated Planning and Analysis of Wide Area Measurement and Protection Systems

Halil Alper Tokel[1], Rana Al Halaseh, Gholamreza Alirezaei, and Rudolf Mathar

***Abstract*—The recent efforts to extend measurement and communication infrastructures in power grids have brought about new potentials for a more efficient and optimized grid operation using the data which become available. Due to this integration, on the other hand, not only the network expansion and planning but also the development of new applications must be carried out under consideration of the interdependencies between the power grid and the underlying communication network. 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. In the first part of the paper, we investigate the performance of PMU-based state estimation techniques under several power system and communication network scenarios. In the second part of the paper, furthermore, the presented setup is extended for co-simulative investigations of the machine learning-based fault detection and classification techniques in power systems. The presented results not only validate the developed software solution for similar research studies but also scrutinize the impact of communication network performance on power system state estimation and PMU-based decision-making for fault detection and classification.**

***Index Terms*—cosimulation, power system, communication network, machine learning, wide area measurement and protection**

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 power grids, new applications and concepts have arisen, such as demand side management or virtual power plants [3], along with new system monitoring, situational awareness, and protection applications which also became possible thanks to the data made available by the phasor measurement units (PMU). On the one hand, these developments pave the way for the successful integration of renewables and a more efficient grid operation. 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 applications. 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 applications which are based on the sensor data.

In our recent work, we proposed a novel optimization model which enables a minimum-cost design of a WAMS with a hierarchical heterogeneous communication network [4]. The proposed approach delivers both the required number and locations of phasor measurement units (PMU) and phasor data concentrators (PDC) 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* [5]. Although planning techniques, such as the one presented in [4], 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, a co-simulation environment for the simulative analysis of integrated WAMS planning approaches has been presented in our another recent work [1]. Apart from the integrated planning and simulation of WAMS, the use of PMU data in various situational awareness and decision-making systems is another recent focus in the power system research, especially regarding the big data analysis techniques, see, for example, [6], [7], and [8]. In this context, we have proposed a novel approach for the training of machine learning (ML)-based fault detection techniques in our another recent contribution [2].

In the current work, we provide a combined and comprehensive presentation of the co-simulation framework presented in [1] along with the extensions of fault detection algorithm as presented in [2]. Furthermore, the results presented in [2] are validated in the co-simulation environment [1]. Therefore, the current work is a further step towards a comprehensive planning and co-simulation of measurement, communication, data analysis, and decision-making functions in an *energy management system* based on our previous contributions in [4], [1],

Manuscript received January 15, 2018; revised January 29, 2018. Date of publication: March 15, 2018.

All authors are with the Institute for Theoretical Information Technology, RWTH Aachen University, 52056, Aachen, Germany (e-mails:tokel[1],alhalaseh,alirezaei,mathar@ti.rwth-aachen.de).

The material in this paper was presented in part at the 2nd International Multidisciplinary Conference on Computer and Energy Science, Split, Croatia, June 2017 [1], and in part at the IEEE Power Energy Society Innovative Smart Grid Technologies Conference - North America, Washington DC, USA, February 2018 [2].

Digital Object Identifier (DOI): 10.24138/jcomss.v14i1.444

[1]Corresponding author.

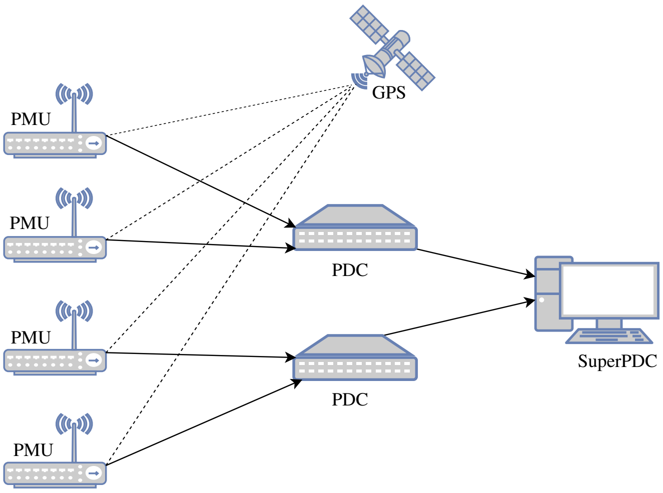

Fig. 1. Hierarchical network architecture of WAMS. PMUs send the phasor measurements, time-stamped by the GPS signal, to a SuperPDC over intermediate PDCs [5].

and [2].

The rest of this paper is structured as follows: We start with a brief note on the details and the operation of a WAMS in Section II, which is followed in Section III by an extended presentation of the co-simulation framework which has already been published in [1]. Particularly, we reveal the details on the extension of open-source communication network simulator OMNeT++ in Section III-B, as we believe that an in-depth understanding of OMNeT++'s software architecture will be a key for interested researchers to minimize the development time and programming effort in similar research studies.

In the second part of the paper, i.e., in Section IV, we extend the framework presented in [1] further for the investigation of wide area protection applications which use synchrophasor measurements. In particular, we integrate external ML libraries in the ICT simulator OMNeT++ and facilitate the co-simulation of a particular fault detection technique, which we proposed in our recent work [2]. We conclude the paper in Section V with a summary and final remarks.

II. WAMS Architecture, Operation, and Planning

A WAMS consists of *i)* PMUs, which measure the voltage and current phasor values available at the system nodes where they are installed, *ii)* several data concentrator units, called PDCs, and *iii)* a data processing center, called SuperPDC (SPDC). *IEEE Standard for Synchrophasor Data Transfer for Power Systems* [5] lays down the architecture for the communication network in a WAMS as shown in Fig. 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 [4]. 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. [9]. We will revisit the function of time alignment in Section IV, where we investigate ML-based fault detection for situational awareness applications.

Due to the increasing importance of WAMS in the operation of power grids, numerous integrated planning approaches have been proposed recently, see for example [10]–[13] and [4] for a comprehensive discussion. The planning approach, which we presented in [4], reveals $i)$ the required number and exact locations of PMU and PDCs for complete 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. Not only the joint formulation of the optimization problem but also the consideration of multiple communication technologies leads to exhaustion of the cost-saving potentials. For the details of the mathematical optimization model, please refer to [4]. In the next section, we provide an extended description and details of the co-simulation framework which enables the integrated planning of a WAMS and its subsequent simulative performance analysis [1].

III. Part I : A Co-simulation Framework for Integrated WAMS Planning and Simulation

The integration of multiple individual domain-specific and powerful simulators as *a cooperative simulation environment (co-simulation)* for power grid simulations has received continuous attention from power system and communication researchers in last two decades. The advantage of a co-simulation is the possibility to benefit from 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 [14], which proposed a run-time environment bridging independent power system and communication network simulators. [15] brings forward the idea of *global event-driven co-simulation* for smart grid simulations. Apart from independently developed co-simulation architectures such as the ones in [14] and [15], there are in the meantime generic open-source libraries, which are designed to connect several simulators in a co-simulation setting, such as MOSAIK [16] which is used in this work. Interested readers may refer to [17] and [18] for a broader view and surveys on the co-simulation of power systems and communication networks.

As far as the analysis of WAMS is concerned, 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, [19] uses the global event list approach of [15] to analyze the effects of communication network on the performance of wide area protection. Similarly, [20] presents a test-bed for the investigation of communication link delays on the wide area monitoring, whereas [21] 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.

The key contributions of the current work in the context of co-simulation of WAMS are the consideration and the implementation of the integrated planning approach as a pre-step prior to co-simulation in order to further accelerate the design and analysis process. Therefore, in this section we present the details of the co-simulation environment and tool chain which enables an integrated planning and a subsequent simulative performance analysis of the designed WAMS.

In the following, we first introduce the entire co-simulation environment used in this section, consisting of OMNeT++ [22] as the communication network simulator and MatDyn [23] as the dynamic power system simulator along with the co-simulation interface MOSAIK [16]. In Section III-B, we intend to provide researchers from various fields with a flexible solution to extend OMNeT++ for similar co-simulation or hardware-in-the-loop requirements. In Section III-C1, we use the presented tool chain for the planning and co-simulation of a WAMS, whose topology is obtained with the optimization model in [4] 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 behavior of the state estimators under abrupt changes in the power system state.

A. 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 [22]. 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 is 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 [1], [22].

MatDyn is a free MATLAB based open-source program to perform dynamic analysis of electric power systems [23]. It is based on the power flow and optimal power flow solutions by the steady state power system analysis toolbox MATPOWER [24]. 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 [1]. For the details of the mathematical models used for the generators and the solutions of DAE systems, please refer to [23].

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 are unevenly distributed on the time axis. Fig. 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. [25] provides an overview of several approaches and a comparison between them in terms of run time and scalability. In [15], 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 independently for a certain step size and then facilitate the required data exchange between the simulators at the end of each step as adopted by the tools EPOCHS [14] and MOSAIK [16]. This approach reduces the programming effort to create a run-time infrastructure for the management of the events. Furthermore, it 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 work, we use the tool MOSAIK due to its generic structure, convenience, and the available detailed documentation [26].

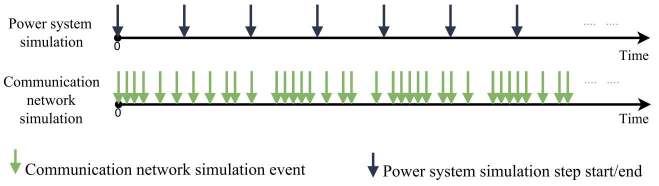
Power system simulation
0
Time
Communication network simulation
0
Time
Communication network simulation event
Power system simulation step start/end

Fig. 2. The advance of simulation time in dynamic power system simulator MatDyn and discrete-even communication network simulator OMNeT++ [1]

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 sub-section.

B. Extension of OMNeT++

The modular architecture of OMNeT++ is depicted in Fig. 3 along with the different libraries of OMNeT++ simulation environment and their relations. In Fig. 3, the *model component library* contains definitions of modules, channels, and classes along with their implementation which can be used

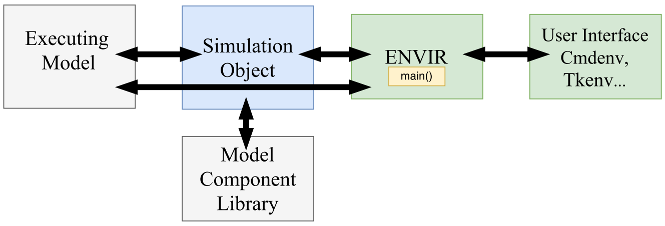

Fig. 3. Modular software architecture of OMNeT++, adapted from [28].

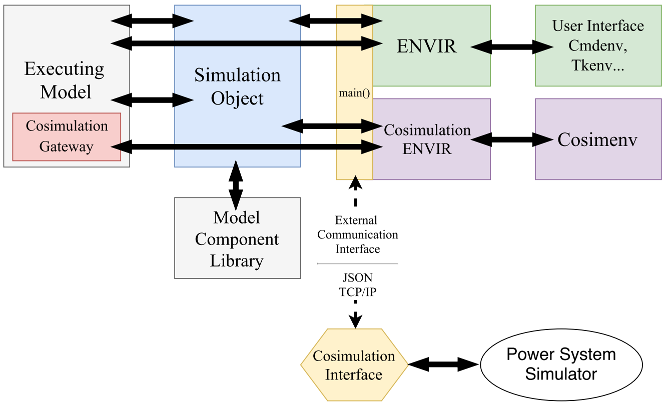

Fig. 4. Extended modular software architecture of OMNeT++. A co-simulation environment and a co-simulation user interface class have been implemented, along with the external communication interface over TCP sockets, extended based on [28], [1].

in a simulation, and *executing model* represents the class instances which are created in a 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 Fig. 4. The extensions in the *main()* function enable the user to provide external simulation settings from MOSAIK so that the simulation can be started with different parameters, input topologies and design requirements without any change in the *.ini* files. On the other hand, the new simulation environment class *cosimulation envir* and the co-simulation user interface `Cosimenv` enable the progress of the simulation in steps with a step size determined by the co-simulation interface MOSAIK in a similar way which was proposed in [27]. 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.

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 `Cosimenv` class to a network module, which we call *co-simulation gateway* and is located in *the executing model* referring to Fig. 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 `Cosimenv`. 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 Fig. 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.

C. 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 [29]. 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 [4], takes place in OMNeT++ with all relevant constraints regarding the telecommunication technologies and the system requirements, using Gurobi 7.0 as the solver [30]. 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++.

1) 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 [31].

We define the system state $\boldsymbol{x}$ of a power grid with n buses by the vector

$$\boldsymbol{x} = [V_{1,re}, \ldots, V_{n,re}, V_{1,im}, \ldots, V_{n,im}] \tag{1}$$

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

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e}, \tag{2}$$

where $\boldsymbol{H}$ is the measurement matrix, which describes the linear relation between the measurement $\boldsymbol{z}$ 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 $\boldsymbol{R} = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ is the covariance matrix, with the variances σ_i^2 of the noise components e_i as its diagonal entries.

2) 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

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad J(\boldsymbol{x}), \tag{3}$$

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 (3) is calculated as

$$\hat{\boldsymbol{x}}_{\text{LWLS},k} = \boldsymbol{G}^{-1}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{z}_k, \tag{4}$$

where 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.

3) 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

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \tag{5a}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{5b}$$

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 process and measurement noises with known covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}$, respectively. Note that $\boldsymbol{w}$ and $\boldsymbol{v}$ 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

$$\tilde{\boldsymbol{e}}_k = \boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k, \tag{6a}$$

$$\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k, \tag{6b}$$

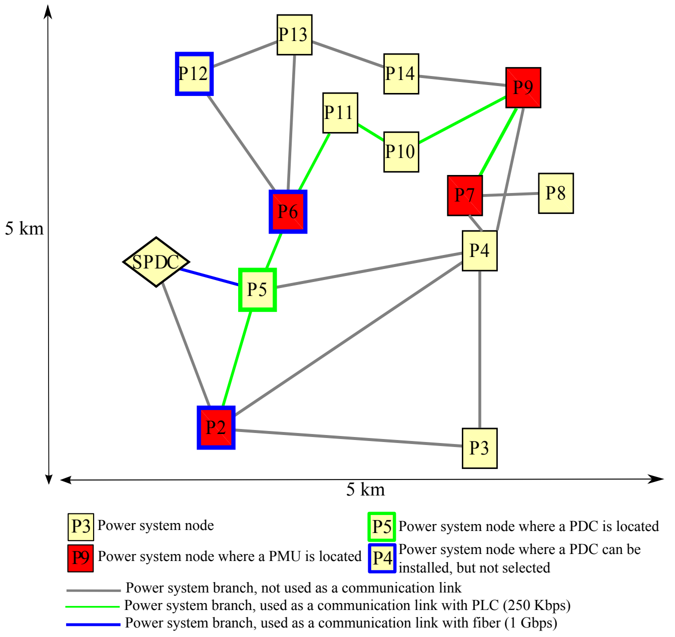
P13
P12
P14
P9
P11
P10
P6
P7
P8
5 km
SPDC
P5
P4
P2
P3
5 km
P3 Power system node
P5 Power system node where a PDC is located
P9 Power system node where a PMU is located
P4 Power system node where a PDC can be installed, but not selected
Power system branch, not used as a communication link
Power system 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 locations of PMUs and PDCs and the optimum communication network topology, obtained by the optimization model in [4]

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

$$\tilde{\boldsymbol{x}}_k = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k, \tag{7a}$$

$$\tilde{\boldsymbol{P}}_k = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q}_{k-1}. \tag{7b}$$

The estimation step then follows as

$$\boldsymbol{K}_k = \tilde{\boldsymbol{P}}_k\boldsymbol{H}^T(\boldsymbol{H}\tilde{\boldsymbol{P}}_k\boldsymbol{H}^T + \boldsymbol{R})^{-1}, \tag{8a}$$

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}\tilde{\boldsymbol{x}}_k), \tag{8b}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\tilde{\boldsymbol{P}}_k, \tag{8c}$$

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 [32] and [31].

4) WAMS Topology and Simulation Parameters: The minimum-cost WAMS topology for the IEEE 14-bus test network is shown in Fig. 5. Note that the node locations are not included in the power system data [29], but approximately generated in this work on a region of 5 km × 5 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 kbps

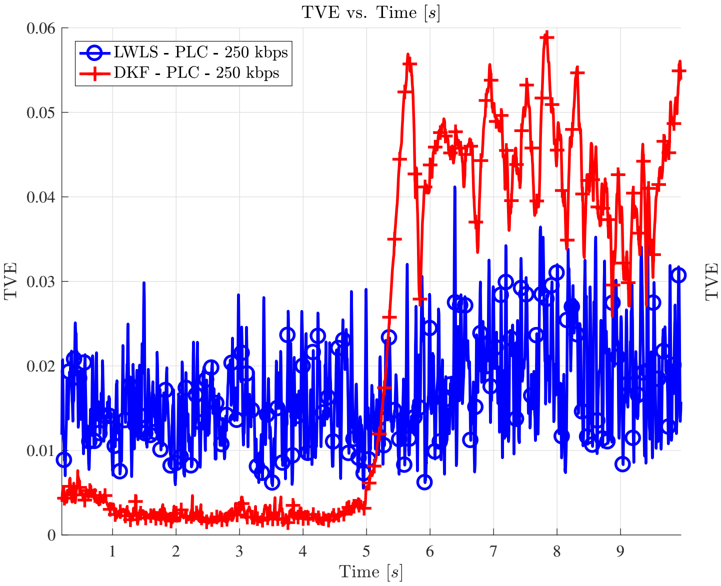

Fig. 6. A sudden load increase of 10% is applied at node P9 at t=5s. DKF performance degrades due to the abrupt change.

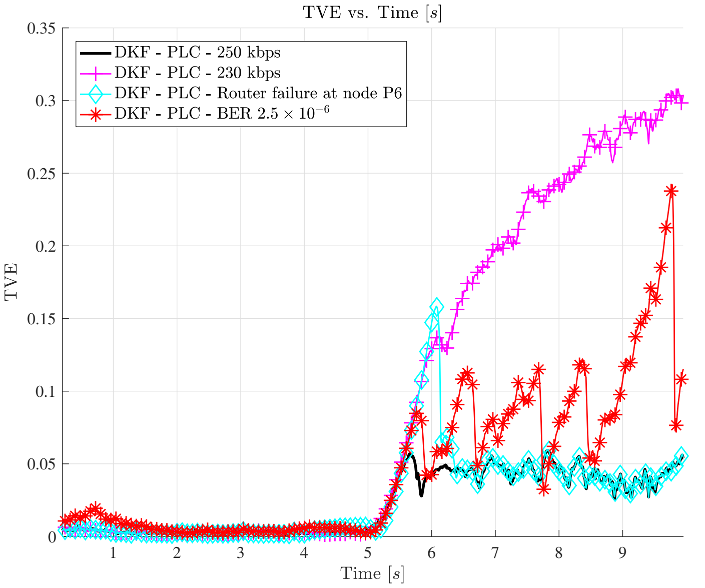

Fig. 7. A sudden load increase of 10% is applied at node P9 at t=5s. Degradations in the communication network performance leads to a deterioration of state estimation accuracy.

and fiber links with a capacity of 1 Gbps. For further cost assumptions, please refer to [4].

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* [5] with a measurement frequency of 50 Hz and the overhead of UDP/IP layers. The step size of the co-simulation and the power system simulation is set to 10 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 dB.

The accuracy of the state estimation at time instance k is assessed with the metric of total vector error (TVE) defined in [5] as

$$\mathrm{TVE}_k = \frac{|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_{\mathrm{true},k}|}{|\boldsymbol{x}_{\mathrm{true},k}|}, \tag{9}$$

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

The simulation duration is set to 10 s during which a sudden load increase of 10% is applied at node P9 at t = 5 s. This scenario is then simulated with the following communication network scenarios :

(i) PLC links with a data rate of 250 kbps, which is used in the optimum topology,
(ii) PLC links with 230 kbps,
(iii) a router failure at node P6 between t = 4 s and t = 6 s, and
(iv) bit error rate (BER) of 1x10^{-6} and 2.5x10^{-4} on PLC links with a data rate of 250 kbps.

In the next subsection, we present and discuss the accuracy of the state estimation observed in the simulations.

5) Results & Discussion: Fig. 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 [31], however, its performance degrades significantly afterwards. Therefore, the comparison and analysis of state estimation techniques, such as recently proposed distributed approaches, must be carried out under several stress scenarios for reliable conclusions.

Moreover, Fig. 7 illustrates the degradation in the DKF state estimation accuracy in the four cases listed in Section III-C4, where communication link performance degradations and router failures are simulated. We observe that in all of the simulated cases, the accuracy of system state estimation deteriorates significantly by the failures and performance degradation in the communication network.

Note that the state estimate at SPDC is the basis for the control and protection 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.

The co-simulation solution presented in this work enables the investigation of communication network impacts on the WAMS operation. Our further aim in the rest of this work is to integrate synchrophasor-based decision-making functionalities in our framework. Therefore, in Section IV, we first briefly introduce the fault detection technique as presented in [2]. Afterwards, we validate the results presented in [2] by using the co-simulation framework.

IV. PART II : ML-BASED FAULT DETECTION AND CLASSIFICATION IN CO-SIMULATION

In Section III, we presented the details of the co-simulation environment with an application of power system state estimation. In this section, the ICT simulator part is further

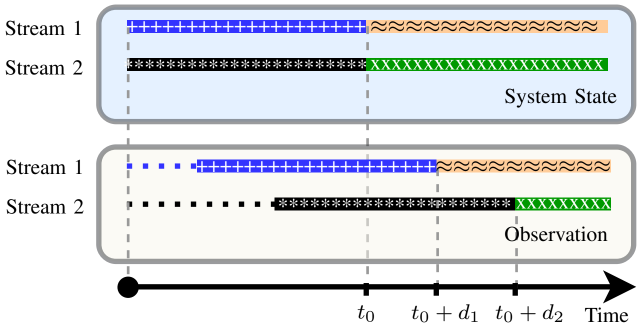

Fig. 8. A possible observation of two PMU streams, which originate from different PMUs. A fault causes detectable changes (illustrated by a change of color) in signal patterns at time t_0, whereas the changes are observed at time $t_0 + d_1$ and $t_0 + d_2$ at SPDC for Stream 1 and 2, respectively, adapted from [2].

extended with the functionalities of data analysis and decision-making for the performance analysis of fault detection and classification technique presented in [2].

The rest of this section is structured as follows: In Section IV-A we deepen our discussion on the operation principle of WAMS regarding the communication delays and situational awareness applications based on PMU measurements, followed in Section IV-B by a note on the recent contributions regarding power system fault detection. In Section IV-C, we summarize the details of the implemented fault detection scheme in [2] for the reader's convenience. In Section IV-D, we present the results of the simulation study by means of the co-simulation framework introduced in Section III.

A. Communication Delays and Decision-Making in WAMS

As introduced in Section II, a WAMS has a hierarchical architecture in which PMU measurements are sent to a PDC and then to the SPDC. As further specified in the related standard for data concentrators [33], for each time stamp, the PDCs align and combine the received measurements in a single packet, which is sent to the SPDC. In this task, the PDC waits for each time stamp a certain amount of time, referred to as *wait time*, before sending the packet. Hence, the measurements experiencing a larger communication delay than the *wait time* will not be available at the SPDC with this packet. Moreover, the standard discusses and allows the deactivation of time alignment functionality for time-critical applications, in which all measurements are forwarded to the SPDC without a *wait time* [2], [33].

Therefore, the observation of the system state in the SPDC will be affected by measurements with different, but known delays. This information is also used by the state estimation function by assigning weights to different measurements based on their delay, for example with a weighted least squares estimator as introduced in Section III-C1, please refer to [34] for a thorough study of power system state estimation. Therefore, the output of the state estimator inherently contains information about the network delays [2].

An illustration of the delayed arrival of measurements is provided in Fig. 8 for two measurement streams which originate from different PMUs. Note that the pattern changes in the signals due to a fault at t_0, which are illustrated by a change of color and background, are observed at the SPDC with delays of d_1 and d_2, respectively. In addition to the available measurement values, the information of the delays is also available due to the time stamps. (Note that the estimation error is not illustrated in Fig. 8.) The central idea of the implemented approach in this section, which we proposed in [2], is to leverage this information in order to detect the system disturbances faster and reliably.

B. Power System Fault Detection and Classification

The faults in power systems can be detected and classified by utilizing the pattern changes in the current and voltage signals in case of a fault [35]. The methods vary from hand-coded expert-defined rules based on certain thresholds to artificial intelligence-based techniques, such as artificial neural networks (ANN), support vector machines, and fuzzy decision systems [6], [36]. Several features and transformations of the signals have been proposed and used for detection purposes like Fourier and wavelet transforms [36].

Although the protection of critical lines and system buses is ensured with local protection equipment like relays and circuit breakers [35], the data made available by PMUs offer the potential to increase the understanding and situational awareness in an energy management center as also proposed in [7] using the output of a PMU-only state estimator for fault detection and classification. Furthermore, the correct and fast detection of the fault location and the identification of the fault type play an important role for the service restoration and the network reconfiguration which are required after a system fault.

In this context, the approaches in [6] and [7] use decision trees, and [8] employs support vector machines for detection of the faults. However, these approaches assume, as discussed above, complete presence of all the measurements in perfect synchronization during the training phase of the ML-based techniques. In the following, we briefly introduce our contribution presented in [2] about the ANN-based fault detection and classification which makes use of training with delayed measurement information for a better performance.

C. ANN-Based Fault Detection and Training Scheme

ANNs, in particular the feed-forward back-propagation multilayer perceptrons (MLPs), have been shown to provide outstanding performance in pattern recognition and classification applications [37]. As indicated in its name, an MLP consists of one or more hidden layers other than the input and the output layers. Each hidden layer has a non-linear activation or transfer function which enables the extraction of the key features in big data sources. This approach is particularly beneficial in problems where a mathematical model for the relation between input and target vectors is difficult to obtain.

An MLP classifier is trained by the supervised back-propagation learning method, where a set of $N \in \mathbb{N}$ labeled input feature vectors, referred to as a training set, is fed into the classifier to iteratively optimize the weights of the neurons

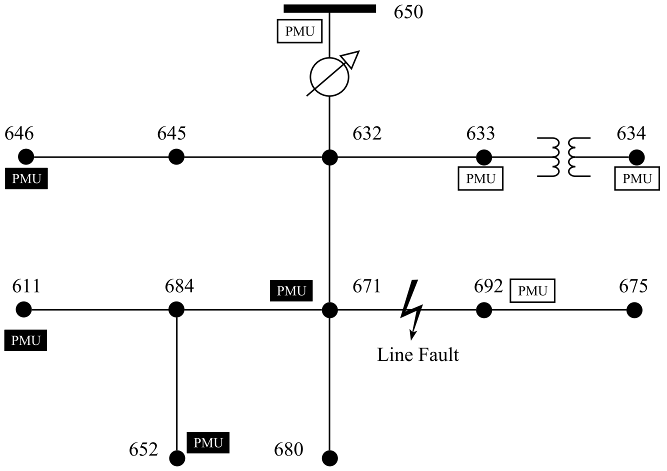

Fig. 9. The topology of the IEEE-13 bus test feeder [38]. A fault detector and classifier for the line between 671 and 692 is implemented and investigated. The PMUs shown in black and white boxes are assumed to send their data to different PDCs. A failure of black PMUs is also simulated.

in its hidden layers. In the current work, the training set $\mathbb{T}$ with N input-target pairs can be written as

$$\mathbb{T} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathbb{R}^M \times \{0,1\}^K \mid n = 1, \ldots, N\}, \tag{10}$$

where $\boldsymbol{x}_n \in \mathbb{R}^M$ is the input feature vector as the state of the power system as introduced in (1), $\boldsymbol{y}_n = [y_{n1}, \ldots, y_{nK}]$ is the target output vector which refers to target class of the corresponding input vector $\boldsymbol{x}_n$, and K denotes the number of possible target classes. The target $\boldsymbol{y}_n$ is an all-zero vector except $y_{nk} = 1$ corresponding to the k^{th} class to which $\boldsymbol{x}_n$ belongs, where $k \in \{1, \ldots, K\}$. Note that, for a fault detection problem, K is equal to 2, referring to the occurrence of a fault or a normal operation. In the current work, the input feature vector $\boldsymbol{x}_n \in \mathbb{R}^M$ is the output of the state estimator which consists of the per unit voltage magnitudes as well as the voltage angles of all the system buses.

In the training of an MLP, the back-propagation algorithm updates the weights via the stochastic gradient descent method such that a certain objective function is iteratively minimized. In this work, the utilized objective function during the training phase is the cross entropy, which is defined for the set $\boldsymbol{\Theta} = \{\mathbf{W}^{(\ell)} \mid \ell = 1, \ldots, L\}$ of weights for $L \in \mathbb{N}$ number of hidden layers as

$$J(\boldsymbol{\Theta}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \log \hat{y}_{nk}(\boldsymbol{\Theta}), \tag{11}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{r_\ell \times r_{\ell-1}}$ is the weight matrix of the hidden layer ℓ, which has r_ℓ neurons, and $\hat{y}_{nk}(\boldsymbol{\Theta})$ is an estimate of y_{nk}, such that $\hat{y}_{nk}(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}, \boldsymbol{x}_n)$. In this content, $f(.)$ denotes the trained MLP model for either fault detection or classification problems. In the current work, MLP detectors have three hidden layers each with 20 neurons. Furthermore, the *softmax* transfer function has been utilized at the output layer, whereas the *hyperbolic tangent sigmoid transfer function* is used as the activator in the hidden layers. For a comprehensive treatment of MLPs, please refer to [37], [39].

The key idea of the detection technique that we proposed in [2] is the training of ML-based algorithms considering the delayed arrival of the measurements. In [2], this approach has been presented and validated by simulated data of the IEEE 13-bus test feeder using the open source distribution system simulator OpenDSS [40] and the neural network toolbox of MATLAB [41], with the following methodology and the parameters:

The generation of the training data for normal and faulty state conditions is carried out in a similar manner which is used in [42] and [43], for faults on the line between the buses 671 and 692. The topology of the test network is shown in Fig. 9. Note that the PMUs shown in black and white boxes are assumed to send their data to different PDCs. The fault resistances are simulated as $1\,\Omega$ and $5\,\Omega$. The normal and faulty state conditions are simulated under various load conditions, where the voltage and current phasor measurements are governed by zero-mean additive white Gaussian noise with an SNR of 30 dB.

In [2], two feed forward ANN fault detectors have been trained with back-propagation. In the first one, the communication delays are not considered, i.e., the output of the state estimator, when noisy synchronized measurements for each time instance are given as the input, is used as the input feature vector and the corresponding system state as the target class, i.e., normal or faulty state. In the second one, on the other hand, the measurements from different PMUs are assigned random communication delays drawn from a uniform distribution with the support $[d_{\min}, d_{\max}]$. The output of the state estimation has been then marked as faulty operation as soon as a measurement value which is recorded after the fault occurrence arrived at the SPDC [2] in order to train the neural network under consideration of delayed arrival of measurements.

In the next sub-section, we present the results regarding the performance of the trained fault detectors in the co-simulation framework.

D. Results in Co-simulation Framework

In the scope of the current work, we have integrated the fault detectors presented in [2] for faults on the line between the buses 671 and 692 in our co-simulation framework. For this purpose, the neural network toolbox of MATLAB is externally integrated to OMNeT++ by using the C API provided by MATLAB. Furthermore, OpenDSS has been connected to MOSAIK through MATLAB using its COM interface [40].

For a performance analysis, we carry out Monte Carlo simulations where the communication delays between a PMU and PDC are drawn uniformly from the interval [10, 70] ms, and the communication delays between a PDC and the SPDC are drawn uniformly from the interval [10, 20] ms in each simulation. The PMUs are set to operate with a reporting rate of 100 frame per second, whereas a state estimation is executed each ms at the SPDC. In the setting of the co-simulation, the state estimates calculated in OMNeT++ are fed into the fault detector in MATLAB which then gives the current operation status as normal or faulty back to the communication simulator OMNeT++.

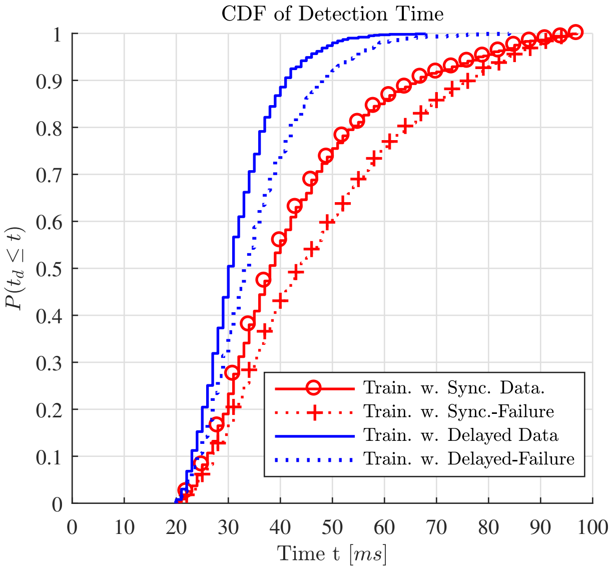

Fig. 10. CDF of the fault detection time in 9000 test cases when the individual PMU-PDC and PDC-SPDC communication delays are uniformly drawn from the intervals [10,70] and [10,20] ms, respectively, in each simulation.

Fig. 10 illustrates the cumulative distribution function (CDF) of the achieved fault *detection time* over all test cases (9000 test cases for each curve) for both considered training methods. We define the *detection time* t_d as the duration between the occurrence of the fault and the detection of the fault by the detector at the SPDC. The curves without markers show the results for the training with delayed data, whereas the curves with markers illustrate the training with the synchronized data. Furthermore, the dotted lines show the results for the cases when the PMU data originating from black PMUs are lost due to a PDC failure, see Fig. 9. Note that the step-wise CDF plots are due to the time step of state estimation, which is equal to 1 ms.

Referring to Fig. 10, we observe a superior performance of the proposed training scheme in the co-simulation framework in alignment with the results presented in [2]. Especially, we observe a considerable improvement in the tail of the distribution, i.e. in the maximum detection time. Therefore, the proposed approach can significantly reduce the detection time. Note that a higher variation of the communication delays of individual PMUs would lead to a more significant improvement in terms of absolute detection time.

Similarly, a superior performance is observed in the case of the loss of PMU data originating from black PMUs due to a PDC failure, see Fig. 9. Therefore, it can be concluded that the proposed approach is more robust in case of failures in the communication network.

V. Conclusion

In this work, we have introduced a co-simulation 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, where linear weighted least squares and discrete Kalman filter techniques are implemented. Furthermore, we have integrated a machine learning-based fault detection algorithm for protection and situational awareness applications. The co-simulation framework presented in this work provides a useful environment for future work in the development and analysis of distributed state estimation, optimization, and fault detection algorithms under consideration of the interdependencies between power grids and communication networks.

Acknowledgment

This work was partly assisted by B.Sc. Thomas Salzmann, Institute for Theoretical Information Technology, RWTH Aachen University. The authors would like to thank him for his effort and commitment.

References

[1] H. A. Tokel, G. Alirezaei, T. Salzmann, and R. Mathar, "Using Co-Simulation for the integrated planning and analysis of wide area measurement systems," in *2nd International Multidisciplinary Conference on Computer and Energy Science 2017 (SpliTech2017)*, Split, Croatia, Jun. 2017.

[2] H. A. Tokel, R. Al Halaseh, G. Alirezaei, and R. Mathar, "A new approach for machine learning-based fault detection and classification in power systems," in *2018 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington DC, USA.

[3] 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.

[4] 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.

[5] "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.

[6] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over PMU data streams," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, Dec 2015, pp. 1–8.

[7] F. Gao, J. S. Thorp, S. Gao, A. Pal, and K. A. Vance, "A voltage phasor based fault-classification method for phasor measurement unit only state estimator output," *Electric Power Components and Systems*, vol. 43, no. 1, pp. 22–31, 2015.

[8] D. Nguyen, S. Wallace, and X. Zhao, "Finding needles in a haystack: Line event detection on smart grid PMU data streams," in *2016 Sixth International Conference on Smart Grids, Green Communications adnd IT Energy-aware Technologies*, 2016, pp. 42–47.

[9] 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.

[10] A. Ghasemkhani, H. Monsef, A. Rahimi-Kian, and A. Anvari-Moghaddam, "Optimal design of a wide area measurement system for improvement of power network monitoring using a dynamic multiobjective shortest path algorithm," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–12, 2015.

[11] M. B. Mohammadi, R. A. Hooshmand, and F. H. Fesharaki, "A new approach for optimal placement of PMUs and their required communication infrastructure in order to minimize the cost of the WAMS," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 84–93, Jan 2016.

[12] M. H. F. Wen and V. O. K. Li, "Optimal phasor data compression unit installation for wide-area measurement systems–an integer linear programming approach," *IEEE Trans. Smart Grid*, vol. PP, no. 99, pp. 1–10, 2015.

[13] H. A. Tokel, G. Alirezaei, S. Baig, and R. Mathar, "An optimization framework for planning of WAMS with a heterogeneous communication network," in *2016 IEEE International Conference on Smart Grid Communications*, Sydney, Australia, Nov. 2016.

[14] 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 Trans. Power Syst.*, vol. 21, no. 2, pp. 548–558, 2006.

[15] 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 Trans. Smart Grid*, vol. 3, no. 3, pp. 1444–1456, Sept 2012.

[16] 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.

[17] K. Mets, J. A. Ojea, and C. Develder, "Combining power and communication network simulation for cost-effective smart grid analysis," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1771–1796, March 2014.

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

[19] 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.

[20] K. Zhu, M. Chenine, and L. Nordstrom, "ICT architecture impact on wide area monitoring and control systems' reliability," *IEEE Trans. Power Del.*, vol. 26, no. 4, pp. 2801–2808, Oct 2011.

[21] 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.

[22] A. Varga, "The OMNeT++ discrete event simulation system," in *Proceedings of European Simulation Multiconference (ESM 2001)*, June.

[23] 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.

[24] 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 Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[25] 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.

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

[27] 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

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

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

[30] Gurobi Optimization Inc., "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[31] S. Sarri, L. Zanni, M. Popovic, J. Y. L. Boudec, and M. Paolone, "Performance assessment of linear state estimators using synchrophasor measurements," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 535–548, March 2016.

[32] 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.

[33] "IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring," *IEEE Std C37.244-2013*, pp. 1–65, May 2013.

[34] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*, ser. Power engineering. CRC Press, 2004.

[35] T. J. D. J. Lewis Blackburn, *Protective Relaying: Principles and Applications*. CRC Press, 2014.

[36] K. Chen, C. Huang, and J. He, "Fault detection, classification and location for transmission lines and distribution systems: A review on the methods," *High Voltage*, vol. 1, no. 1, pp. 25–33, 2016.

[37] S. Haykin, *Neural Networks. A Comprehensive Foundation*, 2nd ed. Pearson Education, 1999.

[38] "Distribution Test Feeders - Distribution Test Feeder Working Group," http://www.ewh.ieee.org/soc/pes/dsacom/testfeeders/index.html, IEEE Power & Energy Society, (Accessed on 04.09.2017).

[39] K. Murphy, *Machine Learning: A Probabilistic Perspective*, 2nd ed. MIT Press, 2012.

[40] R. C. Dugan, "Reference guide: The open distribution system simulator (OpenDSS)," *Electric Power Research Institute, Inc*, vol. 7, 2012.

[41] "MATLAB Neural Network Toolbox Version 9.1," 2017, The MathWorks Inc.

[42] J. Cordova and M. O. Faruque, "Fault location identification in smart distribution networks with distributed generation," in *2015 North American Power Symposium (NAPS)*, Oct 2015, pp. 1–7.

[43] A. Anwar, A. N. Mahmood, and Z. Shah, "A data-driven approach to distinguish cyber-attacks from physical faults in a smart grid," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1811–1814.

Halil Alper Tokel received the B.Sc degree with honors in electrical and electronics engineering in 2011 from Middle East Technical University (METU), Ankara, Turkey, and the M.Sc. degree in electrical engineering and information technology in 2014 from RWTH Aachen University, Germany, both with specialization in telecommunications.

Since September 2014, he has been a Research Assistant at the Institute for Theoretical Information Technology at RWTH Aachen, where he is currently working towards the Ph.D. degree. He has been the recipient of DAAD-TEV Master's Degree Scholarship and the Best Paper Award at the 26th International Telecommunication Networks and Applications Conference. His research interests include integrated planning, optimization, and simulation of energy management systems for future energy grids, machine learning and data analytics with applications in power systems and communication networks.

Rana Al Halaseh received her B.Sc. degree in electrical engineering in 2008 from the University of Jordan, Amman, Jordan, and her M.Sc. degree in 2011 in electrical communication engineering from the University of Kassel, Kassel, Germany.

From 2011 to 2016, she was a Research Assistant at the University of Kassel, and currently she is a Research Assistant at the Institute for Theoretical Information Technology at RWTH Aachen. Her research interests include cognitive radio wireless networks, machine learning algorithms and big data analytics.

Gholamreza Alirezaei received the Diploma and Doctoral degree in electrical engineering from RWTH Aachen University. From 2002 to 2008, he was a research assistant at the Institute of High Frequency Technology of the Faculty of Electrical Engineering and Information Technology at RWTH. From 2008 to 2011, he was the R&D leader at the ATecoM GmbH. From 2011 onwards, he is with the Institute for Theoretical Information Technology at RWTH and he currently works as chief engineer responsible for research and project management. His research interests include the design of communication systems and their development. Furthermore, he tackles hard mathematical problems arisen from estimation, classification and information theory. In particular, nonconvex optimization problems, analytical resource allocation, signal and target classification, and determination of theoretical bounds and inequalities for special functions have been the main topics in the more recent past. Gholamreza has received a number of awards and recognitions. Among others, he is holder of the prestigious Vodafone Young-Researcher Prize, the ITG-Literature Award, and the Friedrich-Wilhelm Prize. He has served as reviewer and member of the Technical Program Committee in numerous IEEE conferences, including ICC, ISIT, ISWCS, GLOBECOM and WiSEE. He also serves as reviewer for both the IEEE Transactions on Signal Processing and the IEEE Transactions on Wireless Communications.

Rudolf Mathar received his Ph.D. in 1981 from RWTH Aachen University. Previous positions include lecturer positions at Augsburg University and at the European Business School. In 1989, he joined the Faculty of Natural Sciences at RWTH Aachen University. He has held the International IBM Chair in Computer Science at Brussels Free University in 1999. In 2004 he was appointed head of the Institute for Theoretical Information Technology in the Faculty of Electrical Engineering and Information Technology at RWTH Aachen University. From 1994 to present he held numerous visiting Professor positions at The University of Melbourne, Canterbury University, Christchurch, Johns Hopkins University, Baltimore, and others. In 2002, he was the recipient of the prestigious Vodafone Innovation Award, and in 2010 he was elected member of the NRW Academy of Sciences and Arts. He is co-founder of two spin-off enterprises. From October 2011 to July 2014 he served as the dean of the Faculty of Electrical and Engineering and Information Technology. In April 2012 he was elected speaker of the board of deans at RWTH Aachen University. Since August 2014 he is serving as Pro-rector for research and structure at RWTH Aachen University. His research interests include information theory, mobile communication systems, particularly optimization, resource allocation and access control.