A hybrid testing module for state estimation in modern power grids based on the OpenPMU and Virtual PMUs

Journal:	*IEEE Open Access Journal of Power and Energy*
Manuscript ID	OAJPE-00092-2021
Manuscript Type:	Regular Paper
Date Submitted by the Author:	14-Sep-2021
Complete List of Authors:	Celeita, David; Universidad del Rosario, School of Engineering, Science and Technology Rios, Mario A.; Universidad de Los Andes, Electrical Engineering ans Electronics Laverty, David; Queen's University Belfast, School of Electronics, Electrical Engineering and Computer Science Forero, Jaime; Universidad de los Andes Facultad de Ingenieria, School of Electrical and Electronic Engineering Jaramillo, Andres; Queen's University Belfast, McLoone, Sean; Queen's University Belfast, School of Electronics, Electrical Engineering and Computer Science
Technical Topic Area:	PMU, real-time simulation
Key Words:	PMU, Power system state estimation, Power system monitoring, Real time systems, Phasor Measurement Units, Real-time simulation

September 10 - 2021

Dear Editor

I have the pleasure of sending you the manuscript entitled ***"A hybrid testing module for state estimation in modern power grids based on the OpenPMU and virtual PMUs"*** authored by: ***David Celeita, Mario Rios, David Laverty, Jaime Forero, Andrés F. Moreno Jaramillo and Sean McLoone*** to be considered for publication as a research paper.

The paper is containing original research and has not been submitted / published earlier in any journal and is not being considered for publication elsewhere. All authors have seen and approved the manuscript and have contributed significantly for the paper.

Ethical Procedure

- The research meets all applicable standards with regard to the ethics of experimentation and research integrity, and the following is being certified/declared true.
- As an expert scientist and along with co-authors of concerned field, the paper has been submitted with full responsibility, following due ethical procedure, and there is no duplicate publication, fraud, plagiarism, or concerns about animal or human experimentation.

A DISCLOSURE / CONFLICT OF INTEREST STATEMENT

- Prof. David Laverty is Associate Editor of the Journal.
- None of the authors of this paper has a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of the paper.
- It is to specifically state that "No Competing interests are at stake and there is No Conflict of Interest" with other people or organizations that could inappropriately influence or bias the content of the paper.

--

David Felipe Celeita Rodríguez Ph.D.
Postdoctoral Research Associate
Department of Electrical and Electronic Engineering
Universidad de los Andes

Professor at School of Engineering, Science and Technology
Universidad del Rosario
ML 751 - Tel 3394949 ext 1746
Bogotá, Colombia

A hybrid testing module for state estimation in modern power grids based on the OpenPMU and Virtual PMUs

David Celeita, *Senior Member, IEEE*, Mario Rios, *Senior Member, IEEE*, David Laverty, *Senior Member*, Jaime Forero *Student Member, IEEE*, Andres F. Moreno Jaramillo, *Student Member, IEEE*, and Seán McLoone, *Senior Member, IEEE*

This paper presents a research context on the virtualization of PMUs and real-time power grids simulation with state estimation. In this research, real-time simulation is introduced to use powerful features for validating state estimation solutions with PMUs. Virtual and online measurement equipment are reviewed in this manuscript to develop an innovative integration of the OpenPMU incorporated with a real-time simulation power grid and additional virtualized PMUs. The implementation of the platform has useful features within the infrastructure that allows the user to reproduce a detailed modeled power grid with simulation software. The use of real-time simulation tools brings several possibilities for improving testing and prototype assessment with higher precision in different applications. In this case, 2 tests power systems are evaluated by realistic integration of IEC61850-9-2 data utilization to observe the performance of a customized state estimation approach. The study implements a versatile methodology for commissioning OpenPMU devices, interacting simultaneously with additional virtual PMUs within the same simulation through Sampled Values to validate the measurement frames and assess the estimation with the generated data. Finally, the proposed work identifies the potential of virtualizing PMUs and the features of the OpenPMU applied to state estimation in conjunction with real-time simulation data.

***Index Terms*—PMU, state estimation, synchrophasor, real-time simulation, measurements, power system monitoring, OpenPMU**

I. Introduction

DIFFERENT efforts have been made in the last decade towards the technological evolution of phasor measurement units (PMUs) as well as improving the data management of such equipment in modern power grids. Such equipment initially dedicated for monitoring the state of an electrical network is crucial for the timely detection of abnormal operating situations in the power system. This allows decision making for protection and control systems in case of failure and to prevent risks in the system's operation. Network monitoring can be done with state estimation models that use information supplied by measurement equipment located only in some nodes of the system [1]. Among the measurement equipment, a phasor measurement unit (PMU) can collect data between 30 and 60 samples per second in 60 Hz systems. The PMU can even record twice the nominal frequency (120 Hz), unlike other measurement equipment with sampling rates between 4 to 6 samples per second [2], [1]. This allows to monitor dynamics of the system that cannot be observed with measurement devices presenting slower reporting times. Therefore, making use of PMU devices in the nodes of the network allows showing dynamics of the system that are not possible to analyze through the use of another measurement device.

The information collected by PMUs to estimate the status of the network contributes to having a better approximation of the current and voltage values in the nodes without measurement devices. This paper presents a scalable simulation platform to assess state estimation solutions with real-time testing, hardware in the loop integration of measurement equipment (OpenPMU), and the use of virtual PMUs with IEC61850-9-2. There are some research tools available such as GridTeractions, a laboratory that allows testing smart and advanced algorithms for modern power grids [3] where many applications are synchronized with an emulated communication network through TCP/IP communication.

This study is going to take part in GridTeractions since it is capable of detailed modeling, to emulate the behavior of real systems. GridTeractions allows communication through a human-machine interface developed on a Raspberry PI 2 (lowercase i) to acquire information from the other available terminals through a local network. The proposed work will implement a platform integrated to GridTeractions developed in a Python script and managed with the Typhoon HIL Control Center program. It allows to run simulations in time series and obtain results in phase domain, to facilitate the user to continuously evaluate the electrical variables. The power grid can be also modeled using Power Factory and then exporting the data via IEC61850-9-2 protocol within the same local network of the laboratory. The following section presents a brief context of virtualization research on monitoring equipment, particularly focused on the PMU. Section III explains the general details of synchrophasor measurement and data communication. Next, in section IV, state estimation on modern power grids is discussed with different approaches. Section V presents the design of the proposed hybrid platform and then validation results are detailed in section VI.

This work was funded by an allocation of the GCRF from the Department for the Economy NI (DfE-GCRF). Andres F. Moreno Jaramillo has also received support from the Department for the Economy (DfE), CREDENCE project (Grant number USI 110).

D. Celeita is with Universidad del Rosario, School of Engineering, Science and Technology, Bogota, Colombia and the Dep. of Electrical and Electronic Engineering Universidad de los Andes (e-mail: df.celeita10@uniandes.edu.co). M. Rios and J. Forero are with the Dep. of Electrical and Electronic Engineering Universidad de los Andes. D. Laverty, A.F. Moreno Jaramillo and S. McLoone are with Queen's University of Belfast, School of Electronics, Electrical Engineering and Computer Science Energy Power and Intelligent Control, Belfast, United Kingdom.

II. The path of virtualizing PMUs and real-time simulation

In this section, a brief compilation of recent contributions on the scope of PMU and PDC virtualization to integrate real-time simulation applications is going to be reviewed. The aim is to identify critical features suitable to improve monitoring and control applications, mainly focused on state estimation using PMUs. Then, the implementation of a proposed design to interface and link an industrial simulation software to an online PMU and manage the data through IEC 61850. This contributes to successfully integrate synchronized physical and virtual measurement equipment to the state estimation module. The proposed platform assessed the compatibility of the OpenPMU integrated to a digital twin build with real-time simulation hardware. The the modular design of OpenPMU [4] lends itself to the virtualization feature, since the blocks are portable between systems, both real hardware and simulated. The validation and testing results showed consistent performance and it is suitable for smart applications and measurement data analysis [5]. Synchrophasor measurements date back to the 1980s, where measurement of two-point phasor variables in a system became a necessity, but insufficient tools existed to do so [1], [6]. Therefore, a first pilot appeared after researching and developing the implementation of a prototype for PMU, with a GPS clock with a precision of one microsecond, (0.021 degrees at 60 Hz). Then, the design of an internal GPS receiver was introduced, 16-bit analog-digital converters for each input of the system, and modem interfaces for remote access to the device. Almost at the same time the IEEE 1344 standard was developed as a precursor to the IEEE C37.118.2 standard. The IEEE 1344 standard aimed to define how the data transmission of PMU devices should be framed with a serial protocol in a packet switch world that could open the discussion with legacy issues. Nowadays the technological boundary that existed back in 1980 is no longer applicable [2], [1]. This is a great opportunity to improve the development of measurement equipment, the connectivity, and data management of multiple PMUs prototypes with better hardware and software design. However, certain challenges are focused on the application to improve monitoring, control, dynamic state estimation, protective schemes [7], cybersecurity and complex solutions for our modern power systems [8], [3]. It has been also proposed the integration of these devices not only at the transmission level but also distribution systems [9], [10] and microgrids [11]. The PMU is critically important for the operation of modern power systems due to the main desired variable when them (the phase angle with high sampling rate), whereas other instruments cannot. Therefore, the PMU applications range is wide as it has been discussed in the literature, and it is useful to diagnose problems in the power system. They are an alternative to monitoring the state of the network as complement to traditional SCADA, since these systems perform non-synchronized measurements normally at a slow scan [1], [6]. However, the measurements of PMU systems are framed to a specific area in the power system in which they are installed, this, due to the synchronization performed between equipment.

III. Synchrophasor measurement and data communication

The standard IEEE C37.188.1 de 2011 defines the PMU as device that produces a synchronized phasor, frequency and frequency change rate of voltage or current signals and a synchronization signal. Therefore, two main parts are identified in the implementation of this device: the processing of the voltage and current signals, and the precise synchronization of time.

A. Measurement and data communication according to Stds IEEE C37.118.1-2011 and IEEE C37.118.2-2011

1) Signal acquisition

The standard initially defines the concept of synchrophasor as the representation in complex components of a sinusoidal signal and an offset angle with respect to a cosine signal that has the rated frequency of the power system and it is synchronized UTC. Eq. 1 shows the notation given by the standard for a sinusoidal signal in a common frame of reference based on [2].

$$x(t) = X_m cos(\omega t + \phi) \tag{1}$$

Subsequently, concepts are defined on the measurement of signals presenting frequencies that are different from the nominal frequency of the system. The example in which the magnitude of the measured signal is constant is used. However, the frequency difference between the sampled signal and the nominal frequency of the system is constant. In Eq. 2 the resultant phasor of a sinusoidal signal with the characteristics mentioned above.

Eq. 2 describes the phasor representation of a sinusoidal signal $x(t)$ as the synchrophasor X, in the case where its magnitude and frequency difference with respect to the nominal frequency of the system is constant.

$$\begin{gathered} X_m(t) = X_M \\ g = f - f_0 = \Delta f \rightarrow X(t) = \left(X_m/\sqrt{2}\right) e^{j(2\pi \Delta f t + \phi)} \end{gathered} \tag{2}$$

The standard defines two types of performance of PMU, type P and type M. As suggested by the guide, the selection between both types is up to the users matching the requirements of particular applications. Some users selected type P for protection solutions, which need a rapid response on the measurement of data and do not require signal filtering [12]. Type M is specified to be used in measurement or in applications that are affected by the aliasing or ”overlap” of signals, which, also need great precision and do not require higher speed in measurement.

The temporal synchronization of the time stamps with PMU devices is carried out by a GPS module, synchronized with the Universal Coordinated Time. The IEEE C37.188.2 2011 standard defines the required data the PMU should deliver for each measurement.

The time stamp delivered by these devices is given in SOC, which is a count of the seconds elapsed since midnight (00:00:00) of UTC from January 1, 1970, until the moment in which the associated measurement is saved. Likewise, the corresponding stamp must be traceable with 1 uS precision within 100 years. Also, it shows how the transmitted information should be, along with its proper sampling rates and data on the quality of the GPS signal. Initially, data should be transferred with predefined reporting rates for frequencies of 10, 25 and 50 frames per cycle for 50 Hz systems and 10, 12, 15, 20, 30 or 60 frames per cycle for systems of 60 Hz. Subsequently, the standard presents the type of data that must be transmitted with the stamps of each measurement, together with their respective meaning. The transferred data contains information about phasor measurements, time stamp and the quality of the received synchronization signal.

IV. State estimation of modern power grids and PMUs

The aim of this work is to design and implement a versatile platform where the user is allowed to reproduce the performance of any power system modeled in PowerFactory. Then, some limited electrical measurements are taken into a real-time platform to extract voltage and/or current signals using virtual PMUs [12] and a physical OpenPMU [13] where Sampled Values (SV) are used as inputs of any state estimator that the user want to assess [14]. The compatibility, accurateness and real-time performance take part of the work as a digital twin. Therefore, a validation of the state estimator is possible, and the user is able to locate the virtual PMUs or the physical OpenPMU with the better convenience (multiple scenarios can be evaluated trough the proposed platform). In this section a brief discussion of the existing state estimators is presented. This study implements one of them to perform a fast validation in two case studies.

A. Weighted least squares (WLS) based state estimation

From the definition of state estimation, the set of measures taken from the system determine the state of each of its nodes to obtain the most probable values of the system from the information and the defined assumptions. By this method, the state estimator is converted into a weighted least squares estimator with the inclusion of the measurement error covariance matrix that serves to weight the precision of each of the measurements [15]. The information and measurements from the physical system model are used to define the equality constraints of the basic weighted least squares optimization and makes this estimator suitable for power systems, particularly with limited number of measurements [16]. The WLS state estimation generates the estimated values with the minimization of the sum of the squares of the errors. The description of the algorithm is given below. Initially, a vector of measurements Z with m number of measurements is defined and a state vector X with the state variables is then considered [17] , such as:

$$Z=\begin{bmatrix} z_1\\ z_2\\ \vdots\\ z_{m-1}\\ z_m \end{bmatrix}, X=\begin{bmatrix} x_1\\ x_2\\ \vdots\\ x_{m-1}\\ x_m \end{bmatrix} \tag{3}$$

For this method, the sets of measurements are used as non-linear functions of the state vector of the system and these functions are denoted in a vector $h_i(x)$:

$$h=\begin{bmatrix} h_1(x_1,x_2,\cdots x_{m-1},x_m)\\ h_2(x_1,x_2,\cdots x_{m-1},x_m)\\ \vdots\\ h_m(x_1,x_2,\cdots x_{m-1},x_m) \end{bmatrix} \tag{4}$$

With these functions, after being evaluated in the real state of the system, they will produce a set of measurements with the real values of the measurements. However, each of these measurements have their own associated unknown error, denoted as vector e:

$$e=\begin{bmatrix} e_1\\ e_2\\ \vdots\\ e_{m-1}\\ e_m \end{bmatrix} \tag{5}$$

Measurement errors are assumed to be independent of each other and they have an expected value equal to zero. With the ideas mentioned above, the complete equation of state can be written as follows:

$$z=[h(x)]+e \tag{6}$$

Then, the solution to the state estimation problem can be formulated as a minimization of the following objective function:

$$J=\sum_{i=1}^{m}\left[\frac{(z_i-h_i(x))^2}{R_i}\right] \tag{7}$$

Which also means,

$$J(X)=[Z-h(x)]^2[R]^{-1}[Z-h(x)] \tag{8}$$

Where R is the covariance matrix of the measurement errors and has a diagonal structure. Each of the diagonal elements is the covariance of its respective measure, and all off-diagonal elements are zero because the measures are assumed to be independent. To find the minimization of this objective function, the derivative must be set to zero. The derivative of the objective function is denoted by $g(x)$:

$$g(x)=\frac{\partial h(x)}{\partial x}=-\left[\frac{\partial h(x)}{\partial x}\right]^T[R]^{-1}[Z-h(x)] \tag{9}$$

To apply this algorithm, the single required information is the measurement error covariance matrix, R, and the measurement function, $h_(x)$. The Jacobian measure, $H(x)$ is simply the derivative of the measure functions with respect to the state vector. The measurement function and measurement Jacobian can be constructed using the known system model, including branch parameters, network topology, and measurement type and locations. The error covariance matrix must also be constructed before the iterations with the precision information of the meters installed in the system.

$$X^{k+1} = X^k + \left[\left[H(X^k) \right]^T [R]^{-1} \left[H(X^k) \right] \right]^{-1} \left[\left[H(X^k) \right]^T [R]^{-1} \left[Z - H(X^k) \right] \right] \tag{10}$$

For the first iteration of the optimization, the measurement function and the measurement Jacobian must be evaluated with a flat voltage profile or start. A flat start refers to a state vector where all voltage magnitudes are 1.0 p.u. and all voltage angles are 0 °. Finally, along with the measurements, the next iteration of the state vector can be calculated over and over again until the desired tolerance is reached. This algorithm is practical when applying dynamic state estimation for power system protection purposes since a detailed model of the system can be introduced in the method [18], [19]. Some other approaches use this method for calibration using PMU data [20]. Although least squares estimation is a well-known and widely used method, robustness could be improved in some applications. Sometimes, it is necessary to perform post-processing of the measurement residuals for the analysis of incorrect data, which can be computationally more expensive. The use of the LAV estimator can compete (computationally) with the WLS estimator. The LAV estimator aims to minimize the sum of the absolute values of the measurement residuals [21].

B. Linear state estimation

During most of the development of traditional state estimators, the idea of accurate simultaneous collection of system-wide measurements was something difficult to achieved [17]. A big assumption that held all traditional state estimation techniques together was that the static state of the power system changed very slowly and operators could afford significant exploration times. Although some current estimators have scan times close to a few seconds, this could take longer for several protection and control applications. PMUs allow the synchronized collection of phasor measurements and, with this technology becoming so prevalent in utilities over the last decades, it is inevitable that it will be used for state estimation applications. This study take advantage not only of PMUs but also the use of open source tools and non-expensive phasor measurement units to validate in a comprehensive real.time platform the performance of a user-defined state estimator. The inclusion of PMU technology in the state estimation can take several forms. PMU measurements can be included using a slightly different formulation than traditional nonlinear weighted least squares or they can be taken into account after a preliminary state of the system has already been determined [22], [23]. Even a small number of these measurements can weigh heavily on the accuracy of the overall state of the system [7]. However, a real application of PMU technology to state estimation would have all the traditional measurements of real and reactive power injections and current and voltage magnitudes replaced by bus voltage phasors and line current phasors. The OpenPMU complies with this condition and, therefore it is going to be implemented with this method. With the usage of PMU measurements some complications can be avoided.

1) Equivalent model π

To understand the fundamental difference between the measurements used in a traditional state estimator and the measurements used in a linear state estimator, it is best to start with the simple two-port π model equivalent for a transmission line. The state of this simple system will be the magnitude of the voltage and the angle at each terminal of the transmission line. Assuming that a PMU is installed at each end of the transmission line, the set of measurements for this system will consist of the voltage phasors of both terminals of the line and the line currents. All values will be considered rectangular because, at the most basic level, this is what the PMU [24] will return. This feature gives the equation of state its linear property. The state of the system is then the following complex vector where V_i corresponds to the voltage at node i of the system and V_j to the voltage at node j of the system.

$$X = \begin{bmatrix} V_i \\ V_j \end{bmatrix} \tag{11}$$

2) Estimation of phasors

The state estimation with phasor measurements is initially developed with the calculation of the cartesian equivalent of the measurement, which is presented below:

$$A\angle\Theta = (A\cos\Theta) + j\,(A\sin\Theta) \tag{12}$$

The set of measurements without errors is considered to be the vertical concatenation of the voltage phasors at each terminal of the transmission line and the line current flows from each end of the transmission line, therefore:

$$X = \begin{bmatrix} V_i \\ V_j \\ I_{ij} \\ I_{ji} \end{bmatrix} \tag{13}$$

The state of the system can clearly be related identically to the voltage measurements on this complex vector. However, the linear relationship between the state of the system and the line flows requires some effort. First, several quantities must be defined. The series admittance and shunt susceptance of the transmission line are as follows:

$$\begin{aligned} Y_{ij} &= (r_{ij} + jx_{ij})^{-1} \\ Y_{i0} &= G_i + jb_i \\ Y_{j0} &= G_j + jb_j \end{aligned} \tag{14}$$

The relationship between the state of the system and the line current flows for the case of a simple transmission line is as follows:

$$\begin{bmatrix} I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} y_{ij} + y_{i0} & -y_{if} \\ -y_{if} & y_{ij} + y_{j0} \end{bmatrix} \begin{bmatrix} V_i \\ V_f \end{bmatrix} \tag{15}$$

Then, the state matrix is completed in such a way that the voltages in each of the nodes that have meters are taken into

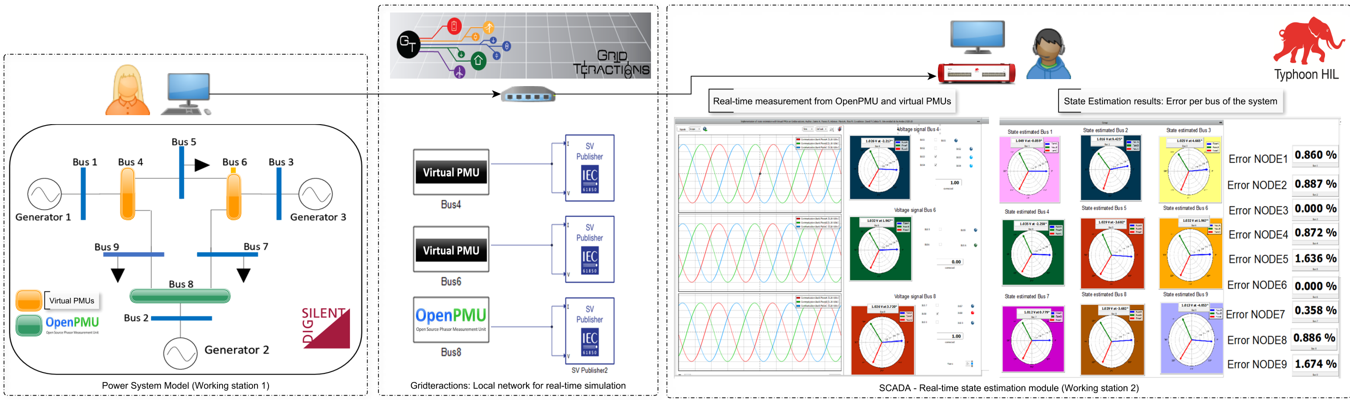

Fig. 1. General diagram of the proposed architecture: Example 9Bus system - Scenario 1

account and, finally the following state equation would be generated:

$$\begin{bmatrix} V_i \\ V_j \\ I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ y_{ij}+y_{i0} & -y_{if} \\ -y_{if} & y_{ij}+y_{j0} \end{bmatrix} \begin{bmatrix} V_i \\ V_f \end{bmatrix} \tag{16}$$

V. Design of the Emulation Framework

The implementation of the linear state estimator integrates a pre-designed power system modeled in PowerFactory that will simulate in a non-interactive mode through a Python script. With this tool, it is possible to do the corresponding tests to verify that the state estimator complies with the necessary parameters to be integrated in the platform. The power system modeled in PowerFactory are 6-bus and 9-bus standard cases. The main elements of the proposed platform are listed as follows:

- One computer (working station 1 in figure 1) with the power system simulation software. The user can load any power system previously modeled with PowerFactory.
- A local network where the virtual PMUs and the OpenPMU will be connected. In this case the proposed platform is connected to an existing network of a dedicated laboratory for multipurpose simulation in real-time (GridTeractions [25], [26]).
- A script to co-simulate PowerFactory (the case study selected by the user) while running the real-time hardware. The script is shared in Appendix 1.
- Real-time simulator (Typhoon) to integrate a limited set of measurements trough the virtual PMUs and the OpenPMU (using Sampled Values with IEC 61850-9-2)
- A script to model the state estimator: The code recognizes the number of meters of any modeled system, since this value may vary depending on the availability assigned to each node. The status of the nodes where the PMUs are located will be entered into the code as input parameters. Then, it generates a vector with the estimated states for all nodes in the system. The error of the estimation should be computed in several scenarios. The script is shared in Appendix 2.
- A typhoon master project that organizes the SCADA of the platform and the real-time hardware simulation (working station 2).

The SCADA window is divided into three sections. The first section shows the status of measured buses. The user can select which buses are going to be monitored by the OpenPMU and virtual PMUs. Different scenarios are going to be established to assess the performance of the proposed platform for two power systems. Once the user has defined the measured points, the signals are plotted in real-time for voltage, and current at each pre-selected node where a meter was assigned. In addition, next to each signal, a checkbox widget is ready to be selected so that the state estimator takes into account the measurements of that node to make the calculations and generates the results. In figure 1, those components are illustrated in the third module at the left (Working station 2). In the second section the Graph Phasor widget is configured for each of the nodes, where it is possible to see the estimated state and error depending on the state estimation algorithm previously designed by the user. The SCADA also illustrates the single line diagram of the power system and each bus show the estimated state (p.u). Therefore, the proposed platform also works as a digital twin of a control center where the users can reproduce different scenarios in the first working station, and then a second user will see how the state estimator performs in real-time. The scenarios for each case study are summarized in table I.

A. The link of virtual PMUs and the OpenPMU through IEC61850

A python script is used to link the power system simulation model, and the real-sime state estimator in Typhoon using IEC 61850-9-2. A macro widget is available so the user can select the state estimator algorithm on the interface (execution rate of the state estimator: 250ms). This in order to update the signals if there is a change in the network due to a failure. As it was initially proposed on the design of the platform, the data exchange between PMUs and the estimator algorithm should be through a communication that complies with IEC 61850-9-2. Typhoon has available a module to export those measurements packages as SV. This allows the user

TABLE I
TEST SCENARIOS: MEASUREMENT POINTS FOR EACH CASE STUDY

Case	Scenario 1	Scenario 2	Scenario 3	Scenario 4
6-bus system	Virtual PMU: Bus 2 Virtual PMU: Bus 4 OpenPMU: Bus 5	OpenPMU: Bus 2	OpenPMU: Bus 5	Virtual PMU: Bus 2 OpenPMU: Bus 5
9-bus system	Virtual PMU: Bus 4 Vitual PMU: Bus 6 OpenPMU: Bus 8	Virtual PMU: Bus 2 Virtual PMU: Bus 4 OpenPMU: Bus 6	Virtual PMU: Bus 3 Virtual PMU: Bus 4 OpenPMU: Bus 8	Vitual PMU: Bus 1 Virtual PMU: Bus 4 OpenPMU: Bus8

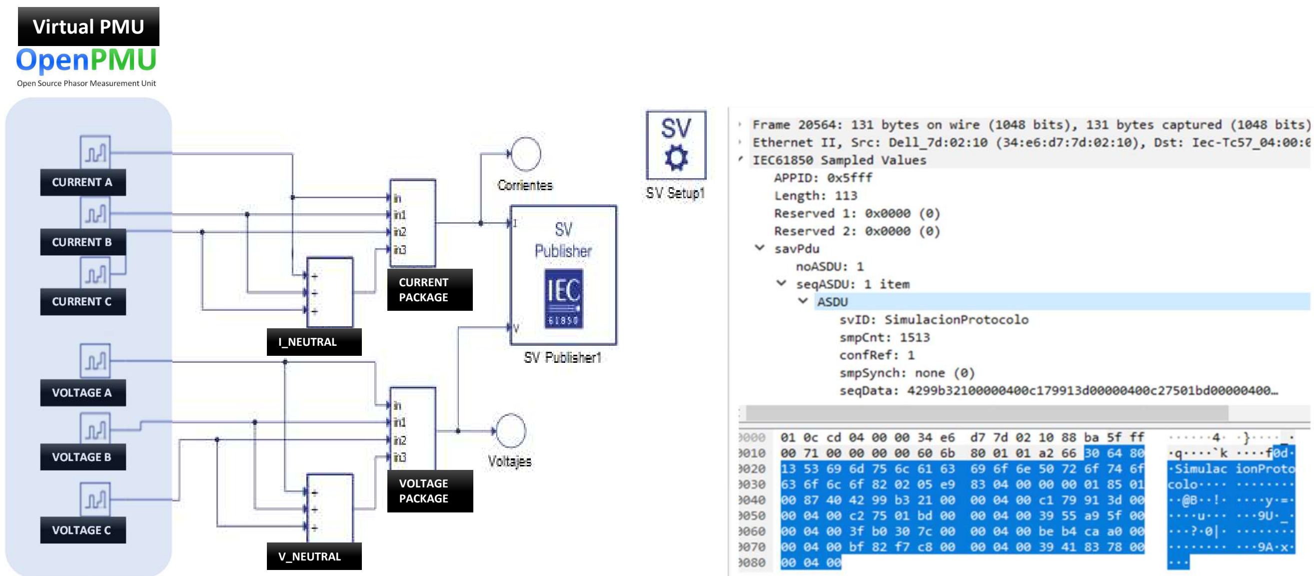

Fig. 2. Signal acquisition from virtual PMUs or OpenPMU and IEC61850-9-2 link of Sampled Values visualized in Wireshark

to establish a Publisher/Subscriber type communication with other hardware or virtual equipment through the Ethernet ports of the real-time equipment. The feature was included to add realism and consistent behavior in the testing stage.

The SV assigns a machine as Publisher and then it sends messages periodically (in this case the virtual PMUs and the OpenPMU). The messages are sent with a description, which allows the subscriber machines selecting and filtering messages where the description matches. Finally, the state estimator receives only messages pre-selected by each machine and analyzes only the information that is addressed to it, voltage and current of the measured buses. In the schematic editor (Typhoon HIL 402) it is possible to find communication modules that allow to implement communication with SV and then the user can capture them using Wireshark.

VI. RESULTS: THE IMPACT OF DATA NOISE AND DELAYS ON STATE ESTIMATION

A. State estimator comparison: 6-bus system

To check the functionality of the platform and versatility to implement different state estimators, three different methods were first implemented in Python. The aim is to compare the performance of each one using the 6-bus system previously modeled in Power Factory. Table II presents the results for each of at each node. After comparing between the 3 state estimation methods and the measurements of each node in the system, the user can consistently assess the error performance and accurateness of the estimation method. One of the main capabilities of the proposed platform is reached, where the user could integrate any state estimator algorithm designed in Python. In this case, the WLS and LAV state estimation methods have both a higher error rate compared to the Linear state estimation. As mentioned in previous chapters, the linear state estimator works with voltage and current phasor measurements, which is why it is compatible with the requirements that were established to integrate it with GridTeractions, virtual PMUs and the OpenPMU.

B. Linear state estimation performance: 6-bus and 9-bus system scenarios

A suitable feature to add scalability to the platform aims that the user can reproduce the perfomance of any other power system previously modeled in PowerFactory. Therefore, in this section, two case study (6-bus and 9-bus standard systems) are evaluated under 4 different scenarios each (refer to Table. I). The user is able to select and check the location of the measurement equipment, virtual PMUs and real hardware (OpenPMU). The user selects the measurement points and the

TABLE II
6-BUS TEST: COMPARISON OF DIFFERENT STATE ESTIMATORS ALGORITHM USING THE PROPOSED PLATFORM

Bus	Real measurements	WLS state estimator	LAV state estimator [21]	Linear State estimator
1	1.0500 0°	1.082 0°	1.0500 0°	1.0565 -0.1183°
2	1.0500 -3.6710°	1.0676 -3.9367°	0.9140 -4.1310°	1.0516 -3.6767°
3	1.0700 -4.2730°	1.0709 -4.3176°	1.0405 -4.2540 °	1.0723 -4.2929°
4	0.9890 -4.1290°	1.0427 -2.9247°	1.0020 -3.8770°	0.9906 -4.1360°
5	0.9850 -5.2760°	1.0103 -5.4631°	0.9980 -5.3410°	0.9864 -5.2234°
6	1.0040 -5.9470°	1.0084 -5.9563°	1.0110 -6.9130°	1.0064 -6.1290°

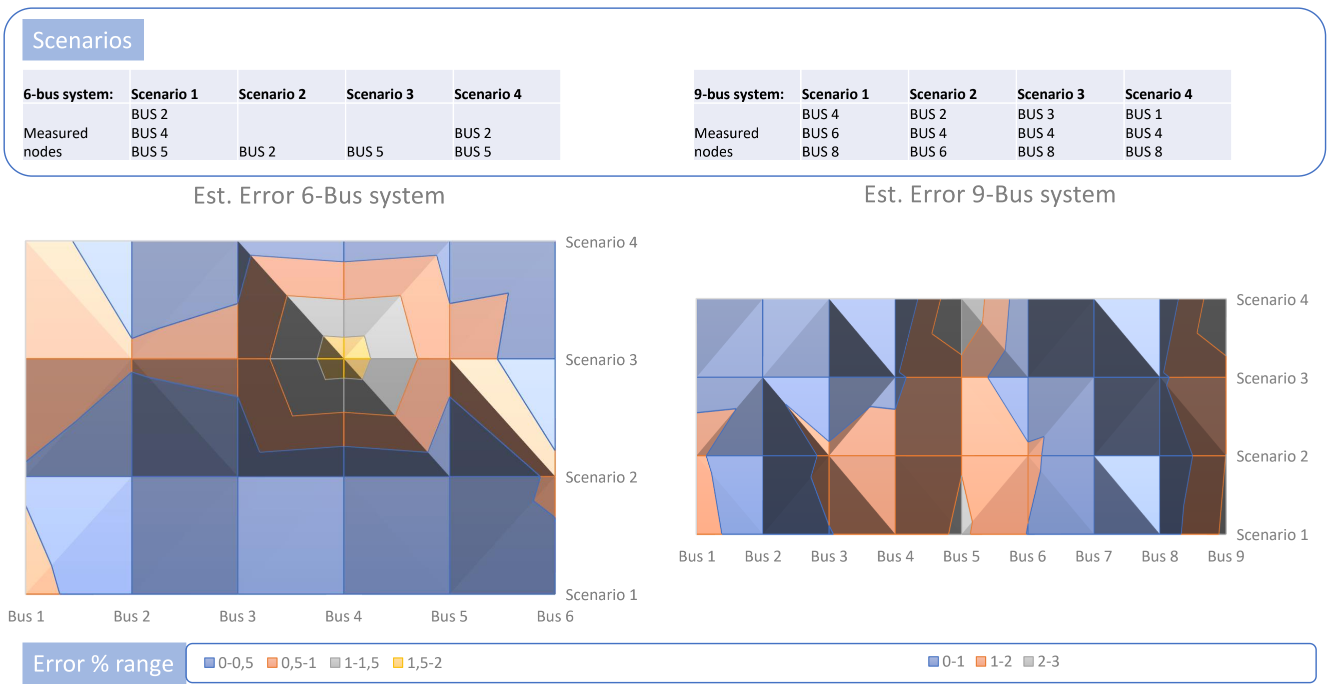

Fig. 3. Linear state estimation performance: 6-bus and 9-bus system for different limited measurement scenarios.

state estimator method, which in this case is the Linear state estimation due to the results in the previous section. Once the user has the testing configuration, results of a digital twin are easily obtained with the proposed platform, with particular focus on the state estimation error.

In figure 3 the error behavior of each tested scenario in both case study (6-bus and 9-bus system) is plotted. Note that in the 6-bus system the user can to select one measurement point (scenarios 2 and 3) where the linear state estimator is capable to reach a consistent solution for each of the non-measured buses. It is important to mention that there is no modeling of noise in the measurement equipment, so the error could be result of communication between the power system simulator and the real-time simulation hardware. Yet, this value is not critical for the estimation since the less accurate estimation is scenario 3 for 6-bus system (note a peak in bus 4 colored in yellow in the plot at the left). It is visible that most of the scenarios and the error values for most of the buses are in the blue range between 0-0.05%.

On the other hand, the results of real-time state estimation for 9-bus system showed a slighly higher error. All the scenarios required at least 3 measured points, where almost all of them performed similar estimation errors (%). It is visible in the right plot of figure 3 that between bus 3 and 6 the error shares the range 1-2%. The worst performance was located at bus 9 for scenario 4 where the error reaches its maximum close to 2.9%. The summary of the error behavior on both power systems when using the linear state estimation method with the real-time simulator and linking SV of the virtual PMUs and the OpenPMU is shown in figure 4.

It is important to note that adding a new case study requires to configure the parameters of impedance, reactance and susceptance in each transmission line. The linear state estimator uses these measurements in its algorithm to give the output values required by the user. The Python code was implemented so that it is possible to change systems without getting deep into these modifications by the user if the power system is simply modeled in PowerFactory. The script is shared on the Appendix so further contributions can be evaluated not only for state estimation but modern challenges

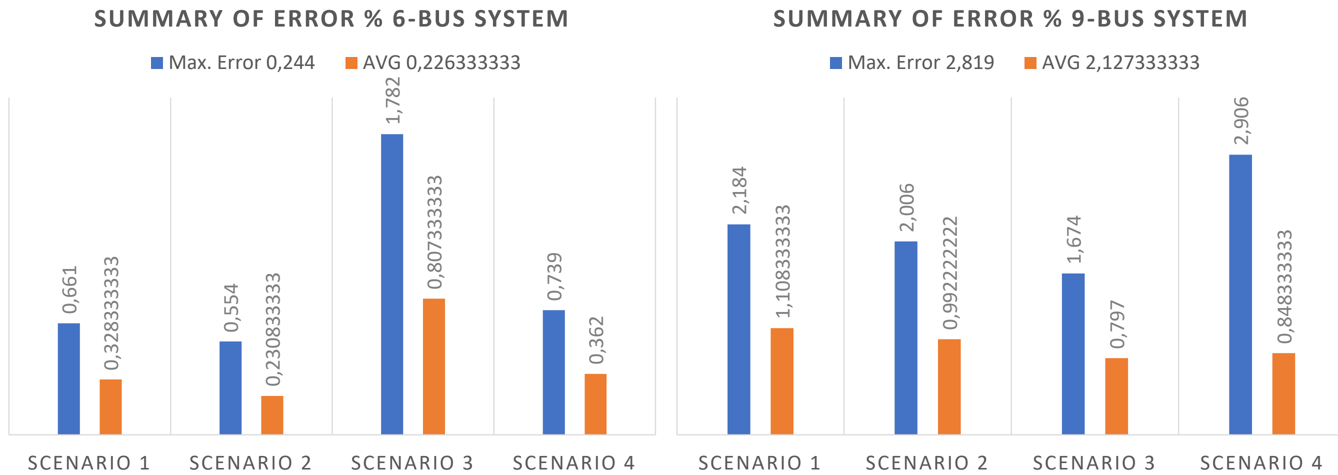

Fig. 4. Max and AVG error summary: Linear state estimation performance using virtual PMUs and OpenPMU with real-time hardware

on data management in power systems.

The two main requirements are crucial for a user to run the hybrid platform: the consistency and stability of pre-modeled systems that can be integrated and the link with GridTeractions. Also among the characteristics of the estimator is that the systems must be implemented in PowerFactory. Since its implementation is done through the Typhoon Hil 402 device, the system conditions must be also compatible with the characteristics of the device. In general, most of the monitoring equipment, relays, PMU based devices and virtual monitoring/protection devices are compatible with this platform because it integrates and centralizes the standardized protocol IEC 61850-9-2 to analyze data in real-time.

The IEC 61850-9-2 compatibility of the platform was integrated to take advantage and build new approaches consistent with the ongoing digitalization of substations. For digital substations, IEC 61850 gives a series of requirements that must be met when purchasing equipment to keep the substation in force over time. Among the requirements found in the standard, the communication protocols are taken into account to develop this hybrid platform for state estimation, since it is seeking to emulate the behavior of the information, particularly the SV which is used in all the stages of the estimator. The communication of the monitoring equipment takes advantage of the local network and the Ethernet ports of the Typhoon HIL 402 where the OpenPMU or any other device that handles the communication type subscriber-publisher, allows reproducing the data interaction between several measured nodes.

VII. Conclusion

This paper presented the design, implementation and validation of a hardware/software platform capable of reproducing the real-time performance to assess state estimation methods in modern power systems. The main features reached on the design allows the user to load a pre-modeled system in PowerFactory. Given the advantages of GridTeractions Laboratory, the user can extract data of certain nodes of the system to connect virtual PMUs and real monitoring equipment such as the OpenPMU, among others. The real-time simulation hardware was the core to centralize the Sampled Values of each measured point and run every 250 ms the customized state estimation algorithm for different monitoring scenarios.

This hybrid platform offers a versatile work environment so the user can focus more on the analysis of data rather single simulation limitations. Advance applied solutions with PMUs and fast prototyping of algorithms are expected due to the capability of validation under a digital twin philosophy. The proposed test scheme was developed to allow the integration of new components and scale the solutions involved with protection, monitoring and control of modern power systems using IEC 61850-9-2.

References

[1] A. G. Phadke, "“modern wide area measurement systems, an overview”," in *8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009)*, 2009, pp. 1–1.

[2] "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, 2011.

[3] J. Zhao, L. Zhan, H. Yin, F. Li, W. Yao, and Y. Liu, "Recent development of frequency estimation methods for future smart grid," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 354–365, 2020.

[4] D. Laverty, G. Egrot, and A. F. Moreno Jaramillo, "Phasor measurement unit development toolkit including pc based polyphase signal generator," in *2021 32nd Irish Signals and Systems Conference (ISSC)*, 2021, pp. 1–5.

[5] C. Huang, C. Thimmisetty, X. Chen, E. Stewart, P. Top, M. Korkali, V. Donde, C. Tong, and L. Min, "Power distribution system synchrophasor measurements with non-gaussian noises: Real-world data testing and analysis," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 223–228, 2021.

[6] A. Phadke and J. S. Thorp, "History and applications of phasor measurements," in *2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 331–335.

[7] M. M. Devi, M. Geethanjali, and A. R. Devi, "Fault localization for transmission lines with optimal phasor measurement units," *Computers Electrical Engineering*, vol. 70, pp. 163–178, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790617307516

[8] Y. Zhang, J. Wang, and J. Liu, "Attack identification and correction for pmu gps spoofing in unbalanced distribution systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 762–773, 2020.

[9] P. M. De Oliveira-De Jesus, N. A. Rodriguez, D. F. Celeita, and G. A. Ramos, "Pmu-based system state estimation for multigrounded distribution systems," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1071–1081, 2021.

[10] S. You, Y. Liu, G. Kou, X. Zhang, W. Yao, Y. Su, S. W. Hadley, and Y. Liu, "Non-invasive identification of inertia distribution change in high renewable systems using distribution level pmu," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1110–1112, 2018.

[11] S. Kumar, N. Das, and S. Islam, "Performance monitoring of a pmu in a microgrid environment based on iec 61850-90-5," in *2016 Australasian Universities Power Engineering Conference (AUPEC)*, 2016, pp. 1–5.

[12] D. Celeita, D. Clavijo, and G. Ramos, "Pmu and pdc server implementation integrated to real-time simulation tools," in *2019 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2019, pp. 1–7.

[13] D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti, and D. J. Morrow, "The openpmu platform for open-source phasor measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 4, pp. 701–709, 2013.

[14] D. M. Laverty, L. Vanfretti, I. Al Khatib, V. K. Applegreen, R. J. Best, and D. J. Morrow, "The openpmu project: Challenges and perspectives," in *2013 IEEE Power Energy Society General Meeting*, 2013, pp. 1–5.

[15] C. Gomez-Quiles, A. de la Villa Jaen, and A. Gomez-Exposito, "A factorized approach to wls state estimation," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1724–1732, 2011.

[16] H. Bilil and H. Gharavi, "Mmse-based analytical estimator for uncertain power system with limited number of measurements," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5236–5247, 2018.

[17] K. D. Jones, J. S. Thorp, and R. M. Gardner, "Three-phase linear state estimation using phasor measurements," in *2013 IEEE Power Energy Society General Meeting*, 2013, pp. 1–5.

[18] J. Zhao, A. Gómez-Expósito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, A. K. Singh, J. Qi, Z. Huang, and A. P. S. Meliopoulos, "Power system dynamic state estimation: Motivations, definitions, methodologies, and future work," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3188–3198, 2019.

[19] A. P. S. Meliopoulos, G. J. Cokkinides, P. Myrda, Y. Liu, R. Fan, L. Sun, R. Huang, and Z. Tan, "Dynamic state estimation-based protection: Status and promise," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 320–330, 2017.

[20] L. Sun, A. Sakis Meliopoulos, Y. Liu, and B. Xie, "Dynamic state estimation based synchronous generator model calibration using pmu data," in *2017 IEEE Power Energy Society General Meeting*, 2017, pp. 1–5.

[21] V. Özdemir and M. Göl, "A robust parameter estimation method based on lav estimator," in *2015 Modern Electric Power Systems (MEPS)*, 2015, pp. 1–5.

[22] M. Zhou, V. Centeno, J. Thorp, and A. Phadke, "An alternative for including phasor measurements in state estimators," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1930–1937, 2006.

[23] G. N. Korres and N. M. Manousakis, "A state estimator including conventional and synchronized phasor measurements," *Computers Electrical Engineering*, vol. 38, no. 2, pp. 294–305, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790611002072

[24] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke, "Synchronized phasor measurement applications in power systems," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 20–27, 2010.

[25] C. Zambrano, C. Trujillo, D. Celeita, M. Hernandez, and G. Ramos, "Gridteractions: Simulation platform to interact with distribution systems," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1–5.

[26] C. Trujillo, C. Zambrano, D. Celeita, M. Hernandez, G. Ramos, J. D. Perez, N. Avendaño, J. Pestana, A. Alba, and C. Cardenas, "An advanced laboratory for research and training of smart grid control and protection schemes," in *2020 IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (I CPS)*, 2020, pp. 1–7.

APPENDIX

Script for linking PowerFactory and Python co-simulation (Example 9-Bus system) with a user-designed state estimator (Example: linear state estimator) using virtual PMUs and OpenPMU measurements trough TYPHOON HIL 402 for real-time testing.

A. Initialization

```
# NOTE: The code specified in this handler will be executed on timer event.
# HIL API is imported as 'hil'
global bus
global linedata9
global VN
global Xest
global Error
global ErrorW

import numpy as np
import math
import cmath
import sys
def ybusppg(num):
    fb = np.array(linedata9)[:,0]      # From bus number...
    tb = np.array(linedata9)[:,1]      # To bus number...
    z = np.array(linedata9)[:,2]       # Z matrix....
    b = np.array(linedata9)[:,3]       # Ground Admittance, B/2...
    a = np.array(linedata9)[:,4]       # Tap setting value..
    y=1/z                               # To get inverse of each element...

    nbus = 9                           # number of buses
    nbrach= len(fb)                     # no. of branches..

    ybus= np.zeros((nbus,nbus),dtype = 'complex_') # Initialise Ybus
    #Formation of the Off Diagonal Elements...
    for k in range(nbrach):
        ybus[int(fb[k])-1,int(tb[k])-1]= ybus[int(fb[k])-1,int(tb[k])-1]-(y[k]/a[k])
        ybus[int(tb[k])-1,int(fb[k])-1]= ybus[int(fb[k])-1,int(tb[k])-1]


    #Formation of the Off Diagonal Elements...
    for m in range(nbus):
        for n in range(nbrach):


            if int(fb[n]-1)==m:
                prueba=y[n]/(a[n]**2)+b[n]
                ybus[m,m]=ybus[m,m]+ prueba

            elif int(tb[n]-1) == m:
                ybus[m,m]=ybus[m,m]+ y[n] + b[n]
    #ybus                 % Bus Admittance Matrix
    #zbus = inv(ybus)     % Bus Impedance Matrix
    return ybus
```

B. Shunt admittance matrix

```
def bbusppg(num): # Line Data for B-Bus (Shunt Admittance)Formation.

    fb = np.array(linedata9)[:,0]          # From bus number...
    tb = np.array(linedata9)[:,1]          # To bus number...
    b = np.array(linedata9)[:,3]           # Ground Admittance, B/2...
    nbus = int(max(max(fb), max(tb)))       # no. of buses...
    nbrach = len(fb)                        # no. of branches...

    bbus = np.zeros((nbus,nbus),dtype = 'complex_')
    for k in range(nbrach):
        bbus[int(fb[k])-1,int(tb[k])-1] = b[k]
        bbus[int(tb[k])-1,int(fb[k])-1] = bbus[int(fb[k])-1,int(tb[k])-1]
    return bbus

#% IEEE - 9...
num=9
ybus = ybusppg(num)                        # Get YBus..
bpq = bbusppg(num)                         # Get B data..
nbus = 9                                  # Get number of buses..
MatrizA=[]
```

C. Virtual PMUs and OpenPMU data arrangement Part 1

```
pmu=[]
pines=[pin1,pin2,pin3]
pinesvalues=[4,6,8]
for i in range(len(pines)):
    if pines[i]==1:
        pmu.append(pinesvalues[i])


for k in range(len(pmu)):
    for i in range(len(linedata9)):
        h4 = np.zeros((nbus))
        if linedata9[i][1]==pmu[k]:
            h4[(linedata9[i][0])-1]=-1
            h4[(linedata9[i][1])-1]=1
            MatrizA.append(h4)
        if linedata9[i][0]==pmu[k]:
            h4[(linedata9[i][0])-1]=1
            h4[(linedata9[i][1])-1]=-1
            MatrizA.append(h4)

Y = np.zeros((len(MatrizA),len(MatrizA)),dtype = 'complex_')
YS = np.zeros((len(MatrizA),nbus),dtype = 'complex_')
```

D. Virtual PMUs and OpenPMU data arrangement Part 2

```
for i in range(len(MatrizA)):
    for k in range(nbus):
        if MatrizA[i][k]==1:
            x=k

        if MatrizA[i][k]==-1:
            y=k
    Y[i,i]=ybus[x,y]
    YS[i,x]=bpq[x,y]

II=[]
for i in range(len(pmu)):
    ii=[]
    for j in range(nbus):
        ii.append(0 if j!=pmu[i]-1 else 1)
    II.append(ii)

M= np.dot(Y,MatrizA)
for a in range(len(MatrizA)):
    for b in range(nbus):
        M[a][b]=M[a][b]+YS[a][b]
II=np.asarray(II)
B=[]
for i in range(II.shape[0]):
    row=[]
    for k in range(II.shape[1]):
        row.append(II[i,k])
    B.append(row)

for i in range(M.shape[0]):
    row=[]
    for k in range(M.shape[1]):
        row.append(M[i,k])
    B.append(row)
B=np.asarray(B)

VN=np.asarray(VN)
```

E. State estimator and error computation

```
Est = np.zeros((len(MatrizA)+(len(pmu))),dtype = 'complex_')
for i in range(len(MatrizA)):
    for k in range(nbus):
        if MatrizA[i][k]==1:
            x=k
        if MatrizA[i][k]==-1:
            y=k
    for t in range(len(linedata9)):
        if (linedata9[t][0]==x+1 or linedata9[t][1]==x+1) and (linedata9[t][0]==y+1 or linedata9[t][1]==y+1):
            f=t

    Est[i+len(pmu)]=(VN[y]-VN[x])/(linedata9[f][2])
for i in range(len(pmu)):
    Est[i]=VN[pmu[i]-1]

W = np.zeros(((len(Est)),len(Est)))
for i in range(len(Est)):
    if i<len(pmu):
        W[i,i]=0.01**2
    else:
        W[i,i]=0.01**2
Wi=np.linalg.inv(W)
Xest=np.dot(np.dot(np.linalg.inv(np.dot(np.transpose(B),B)),np.transpose(B)),Est)

Error=np.zeros((nbus),dtype = 'complex_')
for i in range(len(Xest)):
    Error[i]=100*abs(Xest[i]-VN[i])/VN[i]
    Error[i]= math.sqrt(Error[i].real**2+Error[i].imag**2)
```

David Celeita (SM'2018) received the degree in Electronic Engineering (2011) from the Universidad Distrital, along with an M.Sc. (2014) and a Ph.D (2018) in Electrical Engineering from Universidad de los Andes, Bogota, Colombia. He worked as an automation engineer in low and medium voltage applications, and he was a visiting researcher at Georgia Institute of Technology. He worked as postdoc researcher in the field of protection algorithms for HV lines at CentraleSupélec and an industry partner in France. He is currently a research fellow at Universidad de los Andes and professor at School of Engineering, Science and Technology of Universidad del Rosario. His research interests include Protective - Relaying Control, Smart Grids, Advanced Distribution Automation, Fault Location, and Real-Time Simulation.

Mario A. Ríos received a degree in electrical engineering in 1991 and a M.Sc. Degree in electrical engineering in 1992, both from Universidad de los Andes, Bogotá, Colombia. He received a Ph.D. degree in electrical engineering from INPG-LEG, Grenoble, France, in 1998, and a Doctoral degree in engineering from Universidad de los Andes, in 1998. He worked as a consultant engineer in ConCol (now WSP), Bogotá, Colombia, during 12 years. Also, he was a Research Associate at the University of Manchester (formerly, UMIST). Currently, he is Full Professor at the Department of Electrical Engineering, School of Engineering, Universidad de los Andes, Bogotá.

David M. Laverty (M'10-SM'17) was born in Belfast, Northern Ireland, in 1984. He received the M.Eng and Ph.D. degrees from Queen's University Belfast, Belfast, UK, in 2006 and 2010 respectively. He is currently a Reader with the Energy, Power and Intelligent Control (EPIC) Cluster at Queen's University Belfast, with research interests in anti-islanding detection, telecommunications, cyber-security and Synchrophasor measurement. Dr. Laverty is a senior member of the IEEE and an associate editor for the IEEE OAJPE.

Jaime A. Forero received the B.S degree in electrical and electronic engineering in 2021 from the Universidad de Los Andes, Bogotá, Colombia, where he is currently working toward the master's degree in electrical engineering. He has academic experience in power electronics, control systems analysis, and power quality on distribution networks.

Andres F. Moreno J. was born in Medellin, Colombia, in 1990. He received the B.Eng. degree in Control Engineering from Universidad Nacional de Colombia, and M.Sc. degree in Power Engineering and Sustainable Energy from Swansea University, U.K., in 2012 and 2018 respectively. He is currently doing his 3rd year Ph.D. in Electrical and Electronic Engineering in the Energy, Power, and Intelligent Control (EPIC) Cluster at Queen's University Belfast. His current research interests include integration of distributed energy resources (DER) in the electrical network (photovoltaic systems, wind turbines, electric vehicles, and battery storage). During his research he has focused on the implementation of time-synchronised measurements (OpenPMU), machine learning algorithms and telecommunications to enable online DER identification.

Seán McLoone (S'94–M'96–SM'02) received the M.Eng. degree (Hons.) in electrical and electronic engineering and the Ph.D. degree in control engineering from Queen's University Belfast (QUB), Belfast, U.K., in 1992 and 1996, respectively.,From 1996 to 1997, he was a Post-Doctoral Research Fellow with QUB, where he was a Lecturer from 1998 to 2002. He joined the Department of Electronic Engineering, National University of Ireland Maynooth, Maynooth, Ireland, in 2002, where he served as a Senior Lecturer from 2005 to 2012 and the Head of Department from 2009 to 2012, before returning to QUB in 2013 to take up his current position as a Professor and the Director of the Energy Power and Intelligent Control Research Cluster at the School of Electronics, Electrical Engineering, and Computer Science. His current research interests are in the general area of intelligent systems, with a particular focus on data-based modeling and analysis of dynamical systems. His research has a strong application focus, with many projects undertaken in collaboration with industry in areas such as process monitoring, control and optimization, time-series prediction, and inline sensor characterization.,Prof. McLoone is also a Chartered Engineer and a Fellow of the Institution of Engineering and Technology. He is a Past Chairman of the U.K. and Republic of Ireland Section of the IEEE.