Challenges and Trade-offs of a Cloud Hosted Phasor Measurement Unit-based Linear State Estimator

Vinaya Chakati, Madhurima Pore, Anamitra Pal, Ayan Banerjee, and Sandeep K.S. Gupta
Arizona State University, Tempe, AZ 85281
Email: {vchakati,Madhurima.Pore,Anamitra.Pal,abanerj3,Sandeep.Gupta}@asu.edu

***Abstract*—Being one of the key derivatives of phasor measurement units (PMUs), a synchrophasor-only linear state estimator (LSE) presents a reliable, high quality, and truly dynamic picture of the power grid. However, with the increase in number of buses monitored by PMUs, computational burden will become a critical constraint for the state estimation solver. Although installing additional hardware can be a possible solution, such a solution will considerably raise the cost of capital investment, operation, and maintenance. This paper proposes cloud-computing as a cost-effective alternative to the computational burden problem. This paper also presents feasibility of the cloud based solution with regards to scalability of the system and latency incurred. Our solution is designed to address the critical operational parameters such as latency and variable network sizes. Additionally, the LSE application establishes robust communication procedures to process inputs arriving at high data rates from multiple PMUs. The paper concludes by highlighting future research directions for enhancing such cloud based solutions.**

I. Introduction

The demand for real-time analysis, visualization, and intelligent management of the power grid has given an impetus to the recent developments in the field of PMU based applications [1]. The number of PMUs in North America has doubled over the past 5 years [2]. China is placing PMUs by default in all the substations that they are constructing. It is likely that in the near future the transmission system will be completely monitored by PMUs.

As one of the key derivatives of this technology, a three-phase tracking synchrophasor-only LSE was developed in 2013 for the extra high voltage network of Dominion Virginia Power (DVP), a power utility located in the U.S. [3]. Real-time analysis of PMU data using an LSE solver identified the following challenges: 1. processing of massive amounts of data to meet the operational constraints, 2. adapting to network disruptions, 3. scaling to higher network sizes. LSE solvers are compute intensive applications that require operations to be performed on complex valued matrices. The size of the matrices grow with the number of PMUs added. The data for the LSE solvers is obtained from PMU devices that are geographically distributed across the network. PMUs send data continuously at a very high data rate which need to be aggregated for LSE computation. Processing of such massive data by LSE solvers demand the use of high performance computing platforms. In state of the art system, multiple PMUs are connected to a PDC, which in turn connect to a central control unit through fiber optic cables. The control unit is equipped with computing systems of high capabilities that execute the LSE solver at high speed and gives solutions every 33 ms [3]. In the future however, we are moving towards larger networks and with the advent of micro-PMUs the prospect of having a PMU-like device at the distribution level is becoming a reality. In such cases the network size could be much larger than the current bus systems considered in recent works. As such we need larger computational as well as communicational capacities to enable LSE solver execution within the stipulated time frame of 33 ms. An intuitive solution is the usage of cloud infrastructure.

Cloud hosted LSE solvers for large scale power networks has several challenges: a) although the computational capacity of the cloud is immense, the projected scale at which the number of PMUs in a power network are increasing, may very well nullify the computational advantage provided by a given cloud infrastructure, b) although cloud provides a computational advantage over a central control unit based solution, the computation can take place only when the data is available through the wireless network. Large scale networks result in huge amount of data from each PMU to be transmitted to the cloud. In such cases, communication can be a huge road block towards the successful completion of cloud hosted LSE solver within the stipulated time of 33 ms.

In this paper, we evaluate the advantages and challenges of a cloud hosted LSE solver for large scale power networks. Our proposed solution is a LSE solver which accepts voltage and current data from the PMU devices on the network and performs linear state estimation. We design our solution based on the following assumptions:

1) Each bus of the network has PMUs that measure the voltage of that bus and all line currents that emerge out of that bus.
2) A phasor data concentrator (PDC) is associated with one or more PMUs.
3) Data is measured and sent from each PMU device at the rate of 30 samples per second (= 33ms).

Our solution is a Cloud LSE application implemented in C and is hosted on the cloud server in our university campus. Solution is designed to support two setups: **1-PMU-1-PDC**, where each PMU on the network communicates directly with cloud application to send voltage and current data, **Multi-PMU-1-PDC**, where PDC aggregates data from multiple PMUs and sends it to cloud application. Evaluation of this paper depicts the scalability of LSE application to the number

978-1-5386-2890-4/17/$31.00 ©2017 IEEE

of PMUs present on the network. It also presents the limiting factors in order to set up such cloud based applications for meeting the real-time constraints for larger network sizes. The results show that the proposed implementation can easily handle data from PMUs placed at 30 locations for 1-PMU-1-PDC setup or 57 locations for Multi-PMU-1-PDC setup, while satisfying the 33ms operational constraint.

II. Related Works

The synchrophasor based LSEs accuracy, reliability, and bad data detection and elimination capabilities have already been demonstrated in prior research [3], [4]. However, the LSE solver has to perform matrix multiplications involving complex numbers, especially when the system topology changes. As the size of the system monitored by PMUs increases, this computational complexity will become a limiting factor.

Prior research has tried to address this problem in multiple ways. Rather than centralizing the computation at the control center, researchers have suggested effective approaches for parallel or distributed state estimation [5], [6]. Similarly, in [7], by utilizing the natural divisions present in a power system, the entire network was decomposed into a number of non-overlapping subsections based on geographical distance. Their approach also addressed the associated problem of having a large number of tie-line measurements which significantly impact state estimation results of neighboring buses. In a recent study, Maheshwari et al. proposed a cloud framework to support various power system applications and identified limitations in the standard cloud infrastructure [8]. In this paper, we implement a cloud-hosted linear state estimator, with a special emphasis on scalability.

III. Cloud Hosted Linear State Estimator

A. Linear State Estimation

PMU-based linear state estimation is a well-understood concept. With regards to the pi-equivalent model shown in Figure 1, the concept of an LSE can be expressed as follows [3], [9]:

$$\mathbf{Z} = \begin{bmatrix} V_{meas} \\ I_{meas} \end{bmatrix} = \begin{bmatrix} \mathbf{II} \\ \mathbf{yA} + \mathbf{y}_s \end{bmatrix} x + e. \quad (1)$$

In Eq. 1, phasors V_{meas} and I_{meas} represent the voltage and current synchrophasor measurements, while x represents the complex state vector. The $\mathbf{II}$ matrix is an identity matrix that relates the state vector to the voltage measurements that are obtained directly from PMUs. The $\mathbf{y}$ matrix contains the series admittance information and $\mathbf{y}_s$ matrix contains the shunt admittance information. The $\mathbf{A}$ matrix is an incidence matrix which provides information of the location of the current measurements with respect to the system buses. The system state can now be solved in the least-squares sense as:

$$\hat{x} = (\mathbf{B}^T\mathbf{W}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}^{-1}\mathbf{Z} = \mathbf{MZ}, \quad (2)$$

where, $\mathbf{B} = \begin{bmatrix} \mathbf{II} \\ \mathbf{yA} + \mathbf{y}_s \end{bmatrix}$ and $\mathbf{W}$ is the covariance matrix, which can be assumed to be an identity matrix without any loss

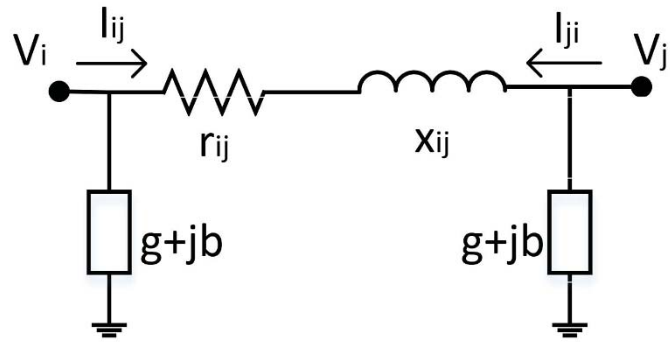


Fig. 1: Two port pi model of a transmission line

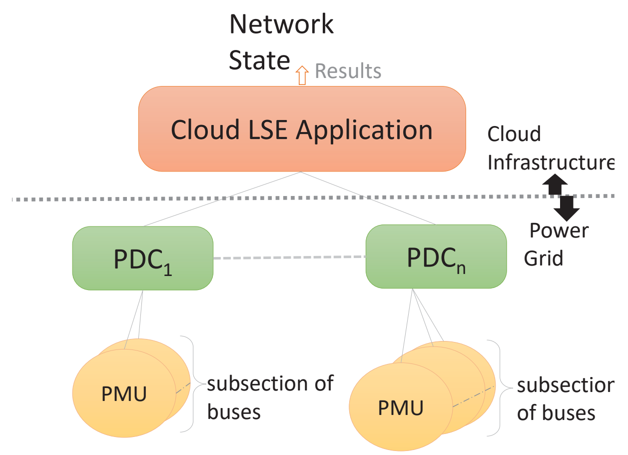


Fig. 2: System model of cloud hosted linear state estimator

of generality. Unlike the traditional SCADA state estimator equations, Eq. 2 is linear and thus no iterations are involved. Matrix $\mathbf{M}$ which involves terms computed using a pseudo inverse, is computed offline and can be saved for real-time use. It is updated only when the system topology changes. Finally, the estimation of the states is performed by matrix multiplication. Therefore, when the LSE is running, performing matrix multiplication and computing matrix inverse consume most of the computation resources.

B. The Cloud Infrastructure

The cloud server hosts the code for the Cloud LSE application. In our case we use a dedicated hardware server for running LSE application that is located in our campus. Run-time data from multiple PMU devices is sent to this server for processing via a PDC. The results from the cloud can be used for performing visual analysis and system monitoring, similar to what is being done in the classical LSE running in DVP.

C. Proposed Framework

In this paper we propose a Cloud LSE application that can provide inputs for management of the power grid based on real-time data obtained from PMUs. As shown in Figure 2 PDCs are intermediate devices that concentrate and aggregate the data from multiple PMUs so that they can be sent as a package to the Cloud LSE application. This application can be configured for different test systems. In our prototype we have focused on increasingly large test systems, while assuming that PMUs are placed at all the buses of the system. We simulate scenarios where massive amounts of real-time data are sent to the cloud application for the estimation of the states. The LSE application computes the network state and reports to the users for live visualization and analysis. The proposed framework deals with real-time processing and communication management of PMU data.

IV. Implementation

This system is implemented as C application with three main functionalities: 1. Handling of input data, 2. Communication management and 3. Linear State Estimation. Each of these functionalities are explained in this section.

A. Input Data Description

The bus configurations of different test systems are used as inputs. PMUs are assumed to be placed on all the buses of these test systems. MATPOWER package from Zimmerman et al. is used to provide the data for different bus configurations i.e. system size [10]. Data obtained from the MATPOWER software includes the network configuration information and voltage and current data for each of the PMU clients.

B. Communication Management

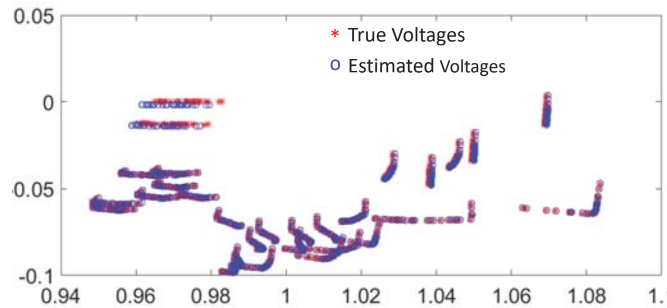
Each PMU sends data to the cloud application /PDC independently. Each bus has a PMU client component that measures bus voltages and associated line currents. 30 such samples are measured every second that need to be sent to the cloud. LSE computation results from the cloud application are made available for further analysis. This client server communication is designed based on C, using TCP/IP sockets. Multiple clients i.e. PMU devices are geographically distributed in the electric grid. We emulate this scenario by client code running on multiple physical machines located in our university campus.

The cloud server is designed to accept connections from multiple PMU devices at a time. Server application uses data queuing for handling continuous data from multiple PMU devices. Performance of the server application is evaluated based on the communication delay and computation time incurred. Communication delay is dependent on various factors such as number of PMU clients, transmission bandwidth, size of data and location of the cloud server.

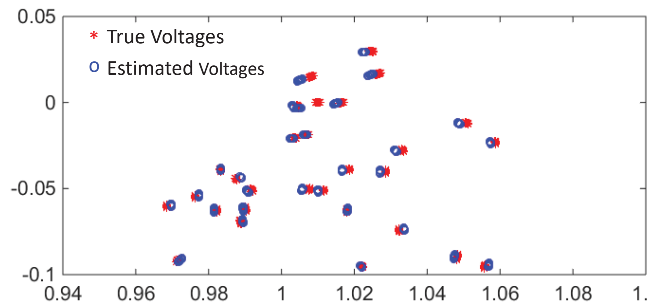
C. Linear State Estimation

The LSE code is a C application running on the cloud server. LSE code runs in two phases, i) *initialization phase*, where it learns about the system parameters, ii) *estimation phase*, where it estimates the states (complex voltages) based on the PMU measurements.

Initialization phase: In the initialization stage, for a specific bus configuration, information about the number of buses, edge connections and line parameters are provided. Line parameter



(a) 30 bus



(b) 57 bus

Fig. 3: Sample accuracy plot of test systems.

information is used as input to compute series $\mathbf{y}$ and shunt admittance $\mathbf{y}_s$. The edge connection information is used for computing $\mathbf{M}$ in equation 2. This phase runs once before it accepts input voltage and current data from the PMU devices.

Estimation phase: In this phase the LSE application accepts the input data from the PMUs and processes it. Voltage and current data is sent from the PMUs at the rate of 30 samples per second. The code is designed to solve Equation 1 and Equation 2 to estimate the bus voltages. In order to solve this equation for complex matrices, we use LAPACK [11] and GSL[12] libraries. The execution time of the LSE depends on the size of the grid, i.e. number of buses. This part of app runs continuously in order to accept a sample of PMU data and process it in real time.

V. Evaluation

Experimental Setup: Every bus in the network is designed to have a PMU client code. Client application reads new values of voltage and current every 33ms and send it to the PDC or cloud server for further processing. Cloud server is configured on an Intel i7-3770K CPU @ 3.50GHz Xeon based server machine located in ASU campus. The server has 4 cores, 8 threads and 16GB RAM. Cloud server is designed to communicate with multiple clients at any time. Cloud LSE application estimates network state for each input sample.

Metrics: The real-time execution of the Cloud app is evaluated based on i) the accuracy of the result, i.e. by comparing the estimated voltages with the actual voltages obtained out of MATPOWER, and ii) the total latency required to obtain the results of each new sample.

TABLE I: Latency results of Cloud LSE application for different test systems.

Test System	No of PDCs	No of Edges	Data Sizes (KB) Input Voltage	Input Current	Line Parameters	Node Info	Time (ms) LSE Computation Time	1-PMU-1-PDC Communication Time	1-PMU-1-PDC Total Time	Multi-PMU-1-PDC Communication Time	Multi-PMU-1-PDC Total Time
14	4	20	0.2	0.2	0.3	0.28	0	8	8	3	3
30	10	41	0.49	0.49	0.47	0.65	4	16	20	10	14
57	19	80	0.92	0.92	1.25	1.39	9	30	39	18	27
118	39	186	1.92	1.92	3.69	4.19	57	62	119	36	93
300	100	411	4.98	4.88	6.68	8.82	96	155	251	92	188

A. Accuracy

The accuracy of the LSE application is a measure of the average difference of the result values as obtained from MATPOWER vs. those obtained from the Cloud LSE application. The morning load pick-up was simulated for the different test systems [13]. Figure 3a shows the plot for the IEEE-30 bus and Figure 3b shows the plot for the IEEE-57 bus test systems, respectively, observed for 30 samples over a duration of 1s. The accuracy observed for results of the LSE application of IEEE-30 and IEEE-57 bus systems are 99.63% and 99.39%, respectively.

B. Latency

In a PMU-based Cloud LSE implementation the PMU data is sent to the cloud for LSE computation and acknowledgment is sent back to the PMU once the results are available. Hence calculating the latency of this entire process is a critical parameter for performance evaluation of the Cloud LSE application. Latency measurement has two components: 1. data transmission time, and 2. data processing time.

1) Communication Latency (t_τ): t_τ is sum of the transmission time incurred for sending the input data from the PMU devices to the Cloud app, t_i, and the time incurred for sending the acknowledgment back to the PMU devices, t_r. Thus, t_τ is given by,

$$t_\tau = t_i + t_r. \quad (3)$$

2) Execution Latency (t_p): If t_p is the time required for processing the input data of voltage and current by the LSE code in order to generate the output.

The total time from the start of sending input data to the time the results are available and acknowledgment is received on the PMU client is given by t_κ as:

$$t_\kappa = t_\tau + t_p. \quad (4)$$

C. Scalability

In this paper, we study the performance of Cloud LSE application for different system configurations. The parameters for configuration of this application are the number of buses in the network and their line connections. As the number of buses increase, the server needs to handle communication with increased number of PMU clients. Moreover, the processing capacity needs to scale up to execute the growing sizes of complex matrices and their computations. Given the real-time constraints of the sample rate of PMU data, the cloud application design needs to address the above mentioned communication and processing demands.

VI. Results

This section shows the results of different test systems for execution time. Results report the configuration details and performance evaluation of different test systems. Input data required for the initialization phase is given by the Line Parameters and Node Information. The data required by the estimation phase is given by the voltage input and current input (refer IV-C). Table I shows the sizes of these data for different test systems. The performance evaluation for a given system is the total time incurred for computation and communication by the Cloud LSE application. Table I shows this latency measurements of one execution run for two setups of the Cloud LSE application.

a) 1-PMU-1-PDC: In this set-up every bus of the network is associated with an individual PMU and a PDC. The PMU-PDC combo communicates directly with the Cloud LSE application to send the current and voltage data.

b) Multi-PMU-1-PDC: In this set-up PDCs are associated with group of PMUs. This set-up represents the configuration where number of PDC devices are approximately 1/3 of the number of buses [14]. Multiple PMUs send data to the PDC which further send it to the cloud.

In Table I, the LSE computation time gives the time required for the LSE solver to generate results once the input from all the PMUs are available in the cloud. The communication time is the time required for the PMU data to reach the cloud application for two cases, one where PMU data is directly sent to cloud and other where PDC aggregate the PMU data and further send it to cloud. The total time for both cases indicates the sum of communication and computation time.

The results above show that the Cloud LSE application can run up to a system size of 30 in 1-PMU-1-PDC set-up, while using the multi-PMU-1-PDC set-up, it can run within the specified time frame (of 33 ms), for a system having a size of 57. The system size corresponds to the number of buses/PMUs. For higher system sizes, the threshold of 33 ms is not met. This time limit is exceeded due to the higher communication delay experienced for sending the individual PMU/PDC data and aggregating that data on the Cloud.

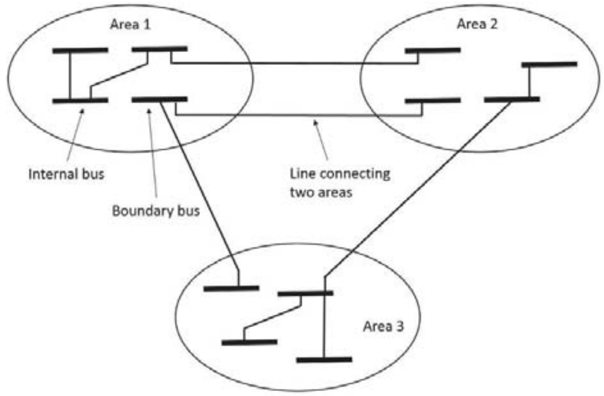


Fig. 4: Power system partitioned into non-overlapping subsystems

VII. Conclusion

This paper investigates the efficacy of running the LSE in the cloud. Proposed solution is designed to support two scenarios, 1-PMU-1-PDC and Multi-PMU-1-PDC. Solution is evaluated for providing the network state estimation within the operational time constraints of the network. The scalability study of our system shows that this system can easily adopted by the networks with size upto 30 bus for 1-PMU-1-PDC setup and upto 57 bus for Multi-PMU-1-PDC set up. For larger network sizes, the problem of high total latency needs to be addressed before this solution can be put into practice. Besides, distributed computing and data confidentiality aspects have not been considered in this paper. Addressing these issues in future research will enable to provide secure and reliable operation.

A. Execution Time for Large Electric Networks

In large distributed network, the cost of communication increases with the size of the network. PDUs are intermediate devices that aggregate and send data to cloud for further processing. Optimal structure of PDC and connected PMU may reduce the communication cost involved in aggregating the data in the cloud server. Designing an optimal configuration to reduce the computation and communication time can be explored in future research in order to run larger sizes of the network.

B. Distributed and Parallel Computing

As the computation for higher system sizes requires longer execution times, there is a need to improve the execution time of LSE application. One approach could be to implement the LSE application in parallel on multicore devices. Also at the server end, each of the samples can be executed on a separate server device. In case of larger system sizes, distributed cluster paradigms such as mapreduce with spark support may be explored for improving the execution speed.

C. Data Privacy and Sharing

As most utilities transition to a more competitive business environment, data privacy becomes a key consideration. As shown in Figure 4, state estimation service and system network partition naturally isolate different electric service providers. This can be ensured by implementing a partitioned LSE (PLSE) in the cloud, similar to what has been proposed for a classical LSE in [15]. Besides data isolation among different end users, data security during transmission between critical infrastructures and external systems should also be considered. SIEGate platform developed under DE-OE0000359 is capable of providing the required secure, low latency data sharing [16]. It also overcomes limitations imposed by frame-based protocols, such as IEEE C37.118, and minimizes the external cyber-attack surface of electric utility control centers. In the future research, the cloud computing platform for LSE can leverage the SIEGate platform and provide secure phasor data exchange with external entities.

Acknowledgment

The authors would like to thank Dr. Kevin Jones and Dr. James Thorp for their valuable inputs.

References

[1] 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.

[2] " U.S. Department of Energy, North American SynchroPhasor Initiative," https://www.naspi.org, 2016, [Online; accessed 5-Dec-2016].

[3] 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*. IEEE, 2013, pp. 1–5.

[4] K. D. Jones, A. Pal, and J. S. Thorp, "Methodology for performing synchrophasor data conditioning and validation," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1121–1130, 2015.

[5] G. N. Korres, "A distributed multiarea state estimation," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 73–84, 2011.

[6] T. Yang, H. Sun, and A. Bose, "Transition to a two-level linear state estimatorpart i: Architecture," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 46–53, 2011.

[7] W. Jiang, V. Vittal, and G. T. Heydt, "A distributed state estimator utilizing synchronized phasor measurements," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 563–571, 2007.

[8] K. Maheshwari, M. Lim, L. Wang, K. Birman, and R. van Renesse, "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*. IEEE, 2013, pp. 1–6.

[9] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. Springer Science & Business Media, 2008.

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

[11] LAPACKTeam, "LAPACKLinear Algebra PACKage," hhttp://www.netlib.org/lapack/, 2016, [Online; accessed 5-Dec-2016].

[12] D. M. Galassi, D. J. Theile, and G. Team, "GSL - GNU Scientific Library," https://www.gnu.org/software/gsl/, 2016, [Online; accessed 5-Dec-2016].

[13] F. Gao, J. S. Thorp, A. Pal, and S. Gao, "Dynamic state prediction based on auto-regressive (ar) model using pmu data," in *2012 IEEE Power and Energy Conference at Illinois*, Feb 2012, pp. 1–5.

[14] 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 Transactions on Smart Grid*, vol. 7, no. 1, pp. 84–93, Jan 2016.

[15] P. Chatterjee, A. Pal, J. S. Thorp, and J. De La Ree, "Partitioned linear state estimation," in *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*. IEEE, 2015, pp. 1–5.

[16] R. Carroll, "SIEgate,Grid Potection Alliance," https://siegate.codeplex.com, 2013, [Online; accessed 5-Dec-2016].