Distributed Estimation of Power System Oscillation Modes under Attacks on GPS Clocks

Yongqiang Wang, *Senior Member, IEEE* João Hespanha, *Fellow, IEEE*

***Abstract*—Phasor Measurement Units (PMU) are playing an increasingly important role in wide-area monitoring and control of power systems. PMUs allow synchronous real-time measurements of voltage, phase angle and frequency from multiple remote locations in the grid, enabled by their ability to align to Global Position System (GPS) clocks. Given that this ability is vulnerable to GPS spoofing attacks, which have been confirmed easy to launch, in this paper we propose a distributed real-time wide-area oscillation estimation approach that is robust to GPS spoofing on PMUs and their associated Phasor Data Concentrators (PDCs). The approach employs the idea of checking update consistency with histories and across distributed nodes and can tolerate up to one third of compromised nodes. It can be implemented in a completely decentralized architecture and in a completely asynchronous way. The effectiveness of the approach is confirmed by numerical simulations of the IEEE 68-bus power system models.**

***Index Terms*—Wide-area monitoring, Synchrophasors, GPS spoofing, cyber attacks, distributed optimization**

I. Introduction

Phasor Measurement Units (PMUs) are widely regarded as one of the most important measurement devices in a power system. By receiving highly accurate time information from the Global Positioning System (GPS), spatially distributed PMUs enable synchronized phasor measurements of voltages and currents from widely dispersed locations in power systems to monitor and control power system dynamics in real-time [1], [2].

However, although the deployment of PMUs steadily increase, several obstacles remain to be overcome before effective *real time* wide-area monitoring and control can be established. First, existing PMUs are organized in a centralized infrastructure, which is not only susceptible to single-point failures, but also subject to computation and communication bottleneck on the central data concentrator. For example, in the Eastern Interconnection of the US grid, all PMU measurements (more than 100,000 data points every second) are sent to a super phasor data concentrator (PDC). As the number of PMUs continuously grows, this centralized structure will not be sustainable, and a distributed cyber-physical architecture has to be constructed instead. Secondly, PMUs rely on GPS signals to synchronize spatially distributed measurements. As recently confirmed in various field tests (e.g., [3]), PMU GPS receivers are susceptible to spoofing attacks, which can deceive a GPS receiver by counterfeit GPS signals and hence disturb the time synchronization process of the receiver. Asynchronized clocks lead to inaccurate time-stamping and hence make measurements inaccurate or even unusable. Unless there is a legitimate way to detect, localize, and mitigate GPS spoofing attacks, they will lead to completely inaccurate monitoring and incorrect control actions.

Recently, there have been a few efforts to address these two problems separately. On the distributed architecture front, [4] proposed a distributed wide-area oscillation monitoring approach based on the Alternating Direction Method of Multipliers (ADMM). Employing ADMM's strength in noise-resilience and strong convergence [5], the proposed approach enables the *online* estimation of oscillation modes through distributed information exchange between phasor data concentrators (PDCs) located at the local control centers and the Independent System Operator (ISO), in comparison to the conventional centralized approaches for oscillation estimation such as the mode metering approach [6], and the Hilbert-Huang transform based approach [7]. On the GPS spoofing attack front, plenty of results have been reported on general spoofing detection (e.g, [8]). Progress has also been made towards evaluating the feasibility and effect of GPS spoofing attacks on PMU measurements and on power system operations in general [9], [10]. However, these results, again, focus mostly on centralized applications, and not on how the outcomes of distributed estimation may be impacted by coordinated GPS spoofing. They also require additional hardware for detection. In this paper we propose a simple variant of distributed ADMM by which one can estimate oscillation modes in the presence of GPS spoofing at multiple PMU locations which can dramatically change or gradually deviate the clocks of PMUs from the actual time. Our detection mechanism is purely algorithmic, and hence does not require extra detection hardware, which is in disparate difference from most existing GPS spoofing detection approaches requiring dedicated GPS receivers or radios (e.g., software-defined radios) to measure and analyze raw GPS signals [11], [12], [13], [14], [15], [16]. Besides detecting spoofing, the proposed approach can also robustly achieve correct oscillation estimation even in the presence of attacks, as long as the number of spoofed PMUs is less than one third of the total number of PMUs, an observation which is reminiscent of the Byzantine general problem [17]. We develop three different algorithms. The first algorithm considers attacks on synchronous estimation of oscillation modes using local PDCs and a central PDC, while the second

This material is based upon work supported by the National Science Foundation under Grants CNS-1329650 and OAC-1738902.

Yongqiang Wang is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 Email: yongqiw@clemson.edu

João Hespanha is with the Department of Electrical and Computer Engineering, the University of California, Santa Barbara, CA 93106 Email: hespanha@ece.ucsb.edu

algorithm extends it to asynchronous communication. The third algorithm involves estimation using direct communication between the local PDCs. We illustrate the effectiveness of all three algorithms using numerical simulations on the IEEE 68-bus power system model. Preliminary results on the first algorithm were recently reported in [18], but the algorithms and illustrations developed in this journal version are much more detailed. The second and third algorithms are new.

PMU based state estimation for power systems is gaining increased popularity [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] and results have also been reported on state estimation in the presence of bad data caused by measurement errors [30], [31], [32], [33], [34] or false-data-injection attacks [35], [36], [37], [38], [39] including GPS spoofing attacks [40], [41]. However, most of the existing results do not apply to a decentralized architecture. Furthermore, to our knowledge, none of these results address distributed oscillation mode estimation in the presence of GPS spoofing.

II. Problem Statement

A. Background

We model a power system as a network of n synchronous generators and n_l load buses. Each synchronous generator is modeled by a second-order swing equation and each load bus is modeled by two algebraic equations for active and reactive power balance. Based on Kron reduction and linearization, and assuming that the mechanical power inputs of all generators are fixed, the differential-algebraic model can be converted into the following completely differential model [42]:

$$\begin{bmatrix} \Delta\dot{\boldsymbol{\delta}}(t) \\ \Delta\dot{\boldsymbol{w}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_{n\times n} & w_s\boldsymbol{I}_n \\ \mathcal{M}^{-1}L & -\mathcal{M}^{-1}\mathcal{D} \end{bmatrix}}_{\boldsymbol{A}} \begin{bmatrix} \Delta\boldsymbol{\delta}(t) \\ \Delta\boldsymbol{w}(t) \end{bmatrix} \quad (1)$$

$$\boldsymbol{y}(t) = \begin{bmatrix} \Delta\theta_1(t), & \ldots, & \Delta\theta_p(t) \end{bmatrix}^T = \boldsymbol{B}\Delta\boldsymbol{\delta}(t) \quad (2)$$

where $\mathcal{M} = \mathrm{diag}(M_1, \ldots, M_n)$ and $\mathcal{D} = \mathrm{diag}(D_1, \ldots, D_n)$, with M_i and D_i denoting the inertia and mechanical damping of generator i, respectively. $\Delta\boldsymbol{\delta} = [\Delta\delta_1, \ldots, \Delta\delta_n]^T$ denotes the small-signal angle deviation, and $\Delta\boldsymbol{w} = [\Delta w_1, \ldots, \Delta w_n]^T$ denotes the small-signal frequency deviation. L denotes coupling in the Kron-reduced form and w_s denotes the synchronous speed of the system. The measurement $\Delta\theta_i(t)$ in $\boldsymbol{y}(t)$ represents a phase-angle-deviation measurement conducted at PMU i. It is a linear combination (governed by matrix $\boldsymbol{B}$) of the elements of $\Delta\boldsymbol{\delta}(t)$. Here we consider phase angle deviations only. The eigenanalysis of $\boldsymbol{A}$ will produce $2n$ eigenvalues $\lambda_i = -\sigma_i \pm j\Omega_i$ with $j = \sqrt{-1}$, which are called oscillation modes. Our aim is to estimate oscillation modes, i.e., the eigenvalues of $\boldsymbol{A}$ from $\boldsymbol{y}(t)$ *in real time* in a distributed way. Next we describe how this can be achieved using ADMM based Prony algorithm.

B. ADMM based wide-area oscillation monitoring

From dynamical systems theory, the $\Delta\theta_i(t)$ $(1 \le i \le p)$ in (2) can be expressed as

$$\Delta\theta_i(t) = \sum_{k=1}^{n} \left(r_{ik} e^{(-\sigma_k + j\Omega_k)t} + r_{ik}^* e^{(-\sigma_k - j\Omega_k)t} \right) \quad (3)$$

where r_{ik} are complex-valued scalar coefficients.

Suppose the measurement is conducted periodically with a small enough measurement period T (according to Nyquist Theorem, the sampling rate $1/T$ needs to be at least twice of the fastest mode to guarantee a faithful reproduction of the signal). Then applying Z-transform to a total of m measurements $y_i(k) \triangleq \Delta\theta_i(kT)$ $(k = 1, 2, \ldots, m)$ yields

$$y_i(Z) = \frac{b_{0k}^i + b_{1k}^i Z^{-1} + b_{2k}^i Z^{-2} + \cdots + b_{2nk}^i Z^{-2n}}{1 + a_1 Z^{-1} + a_2 Z^{-2} + \cdots + a_{2n} Z^{-2n}} \quad (4)$$

where $b_{0k}^i, \cdots, b_{2nk}^i$ and $a_1, \cdots, a_{2n}$ are coefficients.

The oscillation modes, i.e., $\lambda_i = \sigma_i \pm j\Omega_i$ are determined by the solutions to

$$1 + a_1 Z^{-1} + a_2 Z^{-2} + \cdots + a_{2n} Z^{-2n} = 0 \quad (5)$$

which in turn are completely determined by the coefficients $a_1, a_2, \cdots, a_{2n}$. Therefore, the original problem of estimating $\lambda_i = \sigma_i \pm j\Omega_i$ is equivalent to estimating $a_1, a_2, \cdots, a_{2n}$, which can be achieved using the following two-step Prony algorithm [43]:

Step 1: Determine a_1 through a_{2n} through solving

$$\underbrace{\begin{bmatrix} y_i(2n) \\ y_i(2n+1) \\ \vdots \\ y_i(2n+l) \end{bmatrix}}_{\boldsymbol{c}_i} = \underbrace{\begin{bmatrix} y_i(2n-1) & \cdots & y_i(0) \\ y_i(2n) & \cdots & y_i(1) \\ \vdots & \vdots & \vdots \\ y_i(2n+l-1) & \cdots & y_i(l) \end{bmatrix}}_{\boldsymbol{H}_i} \underbrace{\begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_{2n} \end{bmatrix}}_{\boldsymbol{a}} \quad (6)$$

where l is an integer satisfying $2n+l \le m-1$. Concatenating $\boldsymbol{c}_i$ and $\boldsymbol{H}_i$ in (6) for $i = 1, 2, \ldots, p$, one can obtain $\boldsymbol{a}$ by solving a least-squares problem:

$$\min_{\boldsymbol{a}} \|\boldsymbol{\mathcal{H}}\boldsymbol{a} - \boldsymbol{\mathcal{C}}\|^2 \quad (7)$$

where $\boldsymbol{\mathcal{H}}^T = \begin{bmatrix} \boldsymbol{H}_1^T, \ldots, \boldsymbol{H}_p^T \end{bmatrix}$, $\boldsymbol{\mathcal{C}}^T = \begin{bmatrix} \boldsymbol{c}_1^T, \ldots, \boldsymbol{c}_p^T \end{bmatrix}$, and $\| \bullet \|$ denotes the 2-norm.

Step 2: Based on the vector $\boldsymbol{a}$ obtained from *Step 1*, solve (5) to obtain the roots, say denoted by z_i, $i = 1, 2, \ldots, 2n$. The oscillation modes λ_i can be obtained from $\lambda_i = \ln(z_i)/T$.

The above formulation requires a centralized architecture in which all measurements are sent to a central PDC that solves (7) to get λ_i. To obviate the problems with such an architecture (e.g., communication/computation bottleneck and single-point failure), [4] proposed a decentralized architecture which models a power system as a network composed of N utility companies or areas. Each area is equipped with one aggregated PDC (cf. Fig. 1). It is worth noting that here "PDC" is not just a data aggregator and could be any computing agent that can process PMU data. It is also assumed to be synchronized to GPS clocks. These local PDCs (located at every area control center) receive local PMU measurements, run a local least-squares estimation using these measurements to generate a local estimate $\boldsymbol{a}_i$ of the coefficient vector $\boldsymbol{a}$, and then share the estimated values with a central supervisory PDC located at the ISO. Sharing with the central PDC is crucial because each $\boldsymbol{a}_i$ by itself may be insufficient to capture inter-area oscillation modes. Communication via the central PDC can also guarantee that all local PDCs reach the same estimated value. In the distributed architecture, (7) can be

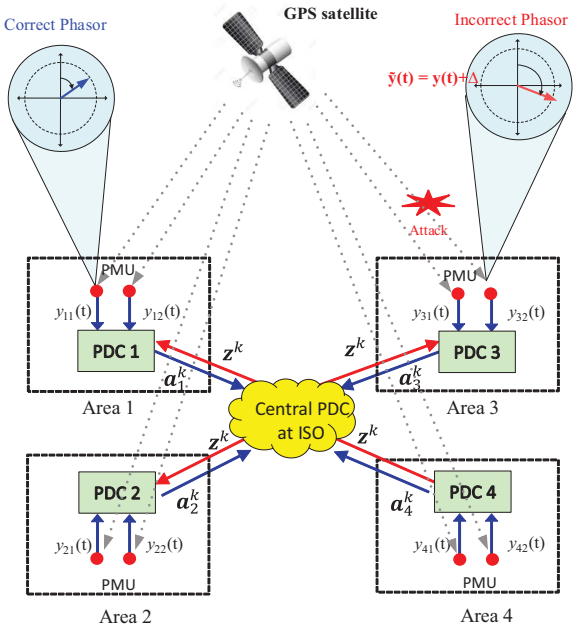

Fig. 1. Distributed architecture for a 4-area power system network. It is worth noting that the presented architecture is just for illustrative purposes, and the approach is applicable to hierarchical PMU-PDC architectures suggested by the synchrophasor standard IEEE C37.118.1 [44]. In fact, in a hierarchical PMU-PDC architecture with local PDCs deployed on different levels, only those PDCs having direct access to PMU measurements need to be involved in the computation.

reformulated as a consensus problem over a network of N local PDCs [4]:

$$\min_{\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_N,\boldsymbol{z}} \sum_{i=1}^{N} \|\hat{\boldsymbol{H}}_i\boldsymbol{a}_i - \hat{\boldsymbol{c}}_i\|^2, \quad \text{s.t.} \quad \boldsymbol{a}_i - \boldsymbol{z} = \boldsymbol{0} \tag{8}$$

for $i = 1, 2, \ldots, N$. Here $\hat{\boldsymbol{H}}_i \triangleq \left[\boldsymbol{H}_{i,1}^T\, \boldsymbol{H}_{i,2}^T \cdots \boldsymbol{H}_{i,N_i}^T\right]^T$ and $\hat{\boldsymbol{c}}_i \triangleq \left[\boldsymbol{c}_{i,1}^T\, \boldsymbol{c}_{i,2}^T \cdots \boldsymbol{c}_{i,N_i}^T\right]^T$ where N_i is the total number of PMUs in area i, and $\boldsymbol{H}_{i,j}$ and $\boldsymbol{c}_{i,j}$ are constructed as in (6) from the time samples of the j^{th} PMU in area i. Parameters $\boldsymbol{a}_i$ are the primal variables for area i. The global consensus solution $\boldsymbol{z}$ is obtained when the local estimates $\boldsymbol{a}_i$ of the N local PDCs ($i = 1, 2, \cdots, N$) reach the same value.

The problem (8) can be solved using ADMM in a distributed way (please refer to [4] for details):

Algorithm 1: Distributed oscillation estimation using ADMM

1) Each local PDC i initializes $\boldsymbol{a}_i^0, \boldsymbol{z}^0$, and $\boldsymbol{w}_i^0$.
2) At iteration k:
a) PDC i constructs $\hat{\boldsymbol{H}}_i^k$ and $\hat{\boldsymbol{c}}_i^k$.
b) PDC i updates $\boldsymbol{a}_i$ according to

$$\boldsymbol{a}_i^{k+1} = ((\hat{\boldsymbol{H}}_i^k)^T\hat{\boldsymbol{H}}_i^k + \rho\boldsymbol{I})^{-1}((\hat{\boldsymbol{H}}_i^k)^T\hat{\boldsymbol{c}}_i^k - \boldsymbol{w}_i^k + \rho\boldsymbol{z}_i^k)$$

and sends $\boldsymbol{a}_i^{k+1}$ to the central PDC. Here $\boldsymbol{w}_i$ denotes the Lagrange multipliers associated with (8) and $\rho > 0$ denotes the penalty factor. ρ makes the Lagrangian differentiable and generally has a large range of permissible values [4].
c) The central PDC calculates

$$\boldsymbol{z}^{k+1} = \sum_{i=1}^{N} \frac{\boldsymbol{a}_i^{k+1}}{N} \tag{9}$$

and broadcasts $\boldsymbol{z}^{k+1}$ to all local PDCs.
d) PDC i updates $\boldsymbol{w}_i$ as $\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k + \rho(\boldsymbol{a}_j^{k+1} - \boldsymbol{z}^{k+1})$.

Since the least-squares problem is convex, as $k \to \infty$, $\boldsymbol{z}^k$ in (9) converges to $\boldsymbol{z}^*$, the optimal solution to the original problem (7) [5]. Moreover, the constraint $\boldsymbol{a}_i = \boldsymbol{z}$ guarantees that all local PDCs get the same $\boldsymbol{a}_1 = \boldsymbol{a}_2 = \cdots = \boldsymbol{a}_N = \boldsymbol{z}^*$. Note that irrespective of the fact whether the PDCs are executing Algorithm 1 in a synchronous or asynchronous fashion, every PDC must be equipped with a clock to keep track of the time-stamping of their respective estimates.

Note that the Prony estimation in (6) is based on the assumption that the incoming disturbance is an impulse function. This assumption is valid for a power system, as testified by the recent IEEE PES technical report on mode estimation [45]. Because of its short-livedness, a fault in a power system can be considered as an impulse function, and the response of the state variables can, therefore, be considered as the impulse response. Hence, after $2n$ sampling instants, the zero dynamics die away completely, and the measurements can be collected in the $\boldsymbol{H}$ and $\boldsymbol{c}$ matrices for solely computing the characteristic polynomial vector $\boldsymbol{a}$, as described above.

C. Problem setup

In the architecture shown in Fig. 1 every PMU and PDC require precise time-stamping, which is achieved through local GPS clocks. Spoofing of any of these clocks will severely disturb the proper execution, convergence, and accuracy of Algorithm 1. Spoofing of PMU clocks, for example, will cause the PMU to generate asynchronous measurements with respect to normal PMUs, leading to phase errors when measurements are correlated. Spoofing of PDC clocks, on the other hand, will lead to failures in keeping track of time-stamping of its estimates, leading to errors when correlated with other normal PDCs. Following [10] which showed that spoofed GPS clocks lead to incorrect oscillation patterns (modes), we model the influence of GPS spoofing as a deviation of $\boldsymbol{a}_i^k$ in Algorithm 1, which corresponds to an oscillation mode error. Although several authentication-based approaches have been proposed (say, e.g., [8]) to defend against external GPS spoofers, an attacker may still be able to compromise the oscillation estimation in Algorithm 1 unless a detection algorithm exists. Our goal is to develop such an algorithm. For convenience we first assume that a central PDC coordinates the computations in the distrusted PDCs, both in synchronous (Sec. III) and asynchronous fashion (Sec. IV). The clock of the central PDC is assumed to be unaffected by spoofing. This assumption will be relaxed in Sec. V where we develop a fully decentralized ADMM resilient to GPS spoofing.

III. Detection using a central PDC and synchronous communications

We first propose an online oscillation estimation approach that is resilient to GPS spoofing attacks in the presence of a central PDC and synchronous communications between PDCs. The basic idea is to enforce certain update restrictions on the local PDCs, i.e., the updated estimation at iteration $k+1$ should be at most ϵ different from iteration k, where ϵ is

a design parameter. This restriction can be imposed simply by adding a saturation function to the update equation for the local estimates. Our goal will be to show that under the proposed framework, the restriction will be automatically satisfied by the central PDC. Therefore, by checking the update status at the central PDC, the ISO can decipher if local PDCs are operating correctly. If the updates from any set of local PDCs violate the update rule, the ISO will receive an alarm that the clocks at these PDCs may have been spoofed.

Algorithm 2: Attack resilient ADMM with a central PDC and synchronous communications

1) Each local PDC i initializes $\boldsymbol{a}_i^0, \boldsymbol{z}^0$, and $\boldsymbol{w}_i^0$.
2) At iteration k:
a) PDC i constructs $\hat{\boldsymbol{H}}_i^k$ and $\hat{\boldsymbol{c}}_i^k$.
b) PDC i updates $\boldsymbol{a}_i$ according to

$$\tilde{\boldsymbol{a}}_i^{k+1} = ((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{H}}_i^k + \rho \boldsymbol{I})^{-1}((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{c}}_i^k - \boldsymbol{w}_i^k + \rho \boldsymbol{z}_i^k)$$

$$\boldsymbol{a}_i^{k+1} = \boldsymbol{a}_i^k + \mathrm{sat}_\epsilon(\tilde{\boldsymbol{a}}_i^{k+1} - \boldsymbol{a}_i^k)$$

and sends $\boldsymbol{a}_i^{k+1}$ to the central PDC. Here ϵ is a positive constant, and $\mathrm{sat}_\epsilon(\bullet)$ is the saturation function defined for every element of a vector as follows:

$$\mathrm{sat}_\epsilon(x) = \begin{cases} \epsilon & \text{if} \quad x > \epsilon \\ x & \text{if} \quad \epsilon \geq x \geq -\epsilon \\ -\epsilon & \text{if} \quad -\epsilon > x \end{cases} \tag{10}$$

c) For $\boldsymbol{a}_i^{k+1}$ received from each local PDC, the central PDC uses the following two rules to detect if it is compromised:

I) ϵ based rule: the central PDC checks for every element of $\boldsymbol{a}_i^{k+1}$ the absolute difference from the corresponding element of $\boldsymbol{a}_i^k$ (the estimate from the same local PDC in the previous iteration). If for some element, the absolute difference is over ϵ, the central PDC determines that the i^{th} PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

II) median based rule: The central PDC checks for every element of $\boldsymbol{a}_i^k$ the absolute difference from the corresponding element in the median vector $\tilde{\boldsymbol{a}}_i^k$ of all PDCs after the transient period (the median is defined on each element as $\tilde{\boldsymbol{a}}_i^k = \mathrm{median}\{\boldsymbol{a}_1^k, \boldsymbol{a}_2^k, \cdots, \boldsymbol{a}_N^k\}$). If for some element, the absolute difference is over δ, then the central PDC determines that the i^{th} PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

Representing the set of all m compromised PDCs as $\mathcal{B}$, the central PDC calculates

$$\boldsymbol{z}^{k+1} = \sum_{1 \leq i \leq N,\, i \notin \mathcal{B}} \frac{\boldsymbol{a}_i^{k+1}}{N - m} \tag{11}$$

and broadcasts $\boldsymbol{z}^{k+1}$ to all local PDCs.
d) PDC i updates $\boldsymbol{w}_i$ as $\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k + \rho(\boldsymbol{a}_j^{k+1} - \boldsymbol{z}^{k+1})$.

The constant ϵ in the above Algorithm 2 is a design parameter, which should be set to a value less than the spoofing-caused deviation, which can be obtained based on published results such as [9] and [10]. It should also be kept secret, otherwise an attacker can try to evade the detection mechanism by judiciously causing deviations smaller than ϵ. ϵ enforces a certain level of consistency between the intermediate results at step k and step $k+1$, and will affect the convergence trajectory, as illustrated in the numerical simulations in Sec. VI. However, it is worth noting that numerical simulations also confirm that ϵ does not affect the final estimate value if set in an appropriate range. More interestingly, we can show that the above "ϵ based rule" can also be used for local PDCs to determine if the central PDC is comprised. More specifically, each local PDC can check if the absolute difference between each element of $\boldsymbol{z}^{k+1}$ and the corresponding element of $\boldsymbol{z}^k$ is over ϵ: If the answer is negative for all elements, then the central PDC is not compromised; otherwise, the central PDC is compromised. The rationale can be easily obtained from the update rule of $\boldsymbol{z}^k$ in (11), which confines $\boldsymbol{z}^{k+1}$ to be within ϵ from $\boldsymbol{z}^k$ if all $\boldsymbol{a}_i^{k+1}$ are within ϵ from $\boldsymbol{a}_i^k$.

The "median based rule" is used to capture attackers which use extremely small deviations to evade the "ϵ based rule". It is motivated by the fact that all $\boldsymbol{a}_i^k$ will converge to the same value after some transient period. (The number of iterations for the transient period can be obtained from simulations or theoretical analysis [5].) That is, when there are no attacks, all $\boldsymbol{a}_i^k$ will converge to the same value, and thus after the transient period every $\boldsymbol{a}_i^k$ will be within a small distance δ from the median $\tilde{\boldsymbol{a}}_i^k$; when a PDC's $\boldsymbol{a}_i^k$ is manipulated by an attacker to gradually deviate from the correct value, it can be detected when the accumulated deviation from the corrected value reaches δ. δ can be set according to the tolerable difference among $\boldsymbol{a}_i^k$ after the transient period. This "median based rule" is also effective in capturing attacks which hold some $\boldsymbol{a}_i^k$ to a constant value, as confirmed by the simulation results in Fig. 14 and Fig. 15.

Remark 1: The ϵ based detection mechanism is affected by neither the model nor the dynamics. As long as every PDC follows the rule correctly, transient events or model uncertainties will not trigger the detection mechanism and cause false alarms, and hence will not be misclassified as attacks.

Remark 2: In the case where IEEE 1588 (PTP) protocol is used to synchronize multiple PMUs and the local PDC in a substation, a GPS spoofing attack on the PTP grandmaster clock with GPS synchronization will affect all PMUs in the substation. Our approach is applicable to this case because it treats a local PDC and all associated PMUs as a single node in the algorithm (also cf. Fig. 1). Of course, the above scenario may lead to the problem of localizing attacks within a substation, which is out the scope of this paper.

IV. Detection using a central PDC and asynchronous communications

In Algorithm 2, all local PDCs are assumed to be able to perform their respective optimization steps with equal speeds, and the communication latencies between the local PDCs and the central PDC are also equal, i.e., the computation and communication among all PDCs are synchronous. However, in reality, different PDCs may not be able to conduct communication and computations in perfect synchronization due

to differences in their processing speeds and heterogenous communication latencies caused by routing and queuing. To address this asynchrony, one can force the central PDC to wait until it receives intermediate communication results from all local PDCs. However, this approach slows down each iteration to accommodate the link subject to the largest end-to-end communication delay, which may lead to unacceptably slow convergence. Fortunately, the recently proposed asynchronous ADMM algorithm provides a way to overcome this issue [46], [47]. Based on [46], [47], we proposed an asynchronous oscillation monitoring algorithm in the absence of GPS attacks [4]. To that algorithm, we now add attack resilience by using a similar crosscheck mechanism as in Algorithm 2. The basic idea of the algorithm is as follows: At each iteration, the central PDC is allowed to use only the more recent results of a subset of the local PDCs to detect attacks and perform updates. Denote the subset of local PDCs as active PDCs and the time instant at which the central PDC updates z^{k+1} as T^{k+1}. The central PDC then broadcasts (z^{k+1}, T^{k+1}) to every local PDC. Upon receiving T^{k+1}, each local PDC i then constructs $\hat{\boldsymbol{H}}_i^{k+1}$ and $\hat{\boldsymbol{c}}_i^{k+1}$ by setting $2n+l$ in (6) to the sample index that is closest to the time instant T^{k+1}. Note that $\Delta\theta(2n+l)$ may not be the most recent measurement sample while constructing the $\hat{\boldsymbol{H}}_i^{k+1}$ and $\hat{\boldsymbol{c}}_i^{k+1}$ matrices. However, to ensure that all PDCs use the same time-window of the measurements to form $\hat{\boldsymbol{H}}_i^{k+1}$ and $\hat{\boldsymbol{c}}_i^{k+1}$, they all use the same value of $2n+l$ as decided globally by the central PDC at every iteration $k+1$. The attack-resilient algorithm can be written as follows:

Algorithm 3: Attack resilient ADMM with asynchronous communications

1) The central PDC initializes T^0 and sends it to all local PDCs.
2) Each local PDC i initializes $\boldsymbol{a}_i^0, \boldsymbol{z}^0$, and $\boldsymbol{w}_i^0$.
3) At iteration k:

a) Given T^k, PDC i constructs $\hat{\boldsymbol{H}}_i^k$ and $\hat{\boldsymbol{c}}_i^k$ using $2n+l$ decided from T^k according to (6).

b) PDC i updates $\boldsymbol{a}_i$ according to

$$\tilde{\boldsymbol{a}}_i^{k+1} = ((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{H}}_i^k + \rho \boldsymbol{I})^{-1}((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{c}}_i^k - \boldsymbol{w}_i^k + \rho \boldsymbol{z}_i^k)$$
$$\boldsymbol{a}_i^{k+1} = \boldsymbol{a}_i^k + \mathrm{sat}_\epsilon(\tilde{\boldsymbol{a}}_i^{k+1} - \boldsymbol{a}_i^k)$$

and sends $\boldsymbol{a}_i^{k+1}$, w_i^k, and k to the central PDC. Here $\mathrm{sat}_\epsilon(\bullet)$ is the saturation function and is defined as in (10) for a given positive constant ϵ.

c) For $\boldsymbol{a}_i^{k+1}$ received from each active local PDC, the central PDC uses the following two rules to detect if it is compromised:

I) ϵ based rule: the central PDC checks for every element of $\boldsymbol{a}_i^{k+1}$ the absolute difference from the corresponding element of $\boldsymbol{a}_i^k$ (the estimate from the same local PDC in the previous iteration). If for some element, the absolute difference is over ϵ, the central PDC determines that the i^{th} PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

II) median based rule: The central PDC checks for every element of $\boldsymbol{a}_i^k$ the absolute difference from the corresponding element in the median vector $\tilde{\boldsymbol{a}}_i^k$ of all PDCs after the transient period (the median is defined on each element as $\tilde{\boldsymbol{a}}_i^k = \mathrm{median}\{\boldsymbol{a}_1^k, \boldsymbol{a}_2^k, \cdots, \boldsymbol{a}_N^k\}$). If for some element, the absolute difference is over δ, then the central PDC determines that the i^{th} PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

Representing the set of all m compromised PDCs as $\mathcal{B}$, the central PDC calculates

$$\boldsymbol{z}^{k+1} = \sum_{1 \le i \le N,\, i \notin \mathcal{B}} \frac{\boldsymbol{a}_i^{k+1}}{N-m} \tag{12}$$

where $\boldsymbol{a}_i^{k+1} = \boldsymbol{a}_i^k$ and $\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k$ for all non-active PDCs whose most recent updates are not available at the central PDC.

d) The central PDC broadcasts $\boldsymbol{z}^{k+1}$, $k+1$, and T^{k+1} to all local PDCs.

f) PDC i updates $\boldsymbol{w}_i$ as $\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k + \rho(\boldsymbol{a}_j^{k+1} - \boldsymbol{z}^{k+1})$ for all active PDCs and as $\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k$ for all non-active PDCs.

In Algorithm 3, the central PDC exchanges the iteration numbers k and $k+1$ with the local PDCs in steps b) and d) to keep track of the order of the received data. Since the iteration number is independent of the absolute time, this additional information exchange (compared with the synchronous communication case) is insensitive to GPS spoofing attacks and will not increase the vulnerability of the system. However, when compared to Algorithm 2, the local PDCs also have to transmit $\boldsymbol{w}$ to the central PDC. This additional information exchange is sensitive to GPS spoofing attacks since $\boldsymbol{w}$ is affected by the absolute time. The above algorithm can guarantee convergence to the global minimum if none of the local PDCs is permanently dormant. In other words, each PDC must be active infinitely often with probability 1 [46]. This ensures that oscillation estimation can be solved in real-time, despite asynchronous updates caused by, e.g., heterogenous computation speeds or heterogenous communication latencies. It is worth noting that if an attack occurs on a dormant node, it cannot be detected by the central PDC until the dormant PDC wakes up and communicates with the central PDC twice consecutively (the detection requires the consistency between two consecutive iterations).

Because in Algorithm 3 the central PDC only uses the data from the subset of active local PDCs that feed measurements to the central PDC fast enough (with small enough communication latencies), there is a trade-off between the number of active PDCs and the lag in oscillation estimation. Given that the number of useable active PDCs correlates positively with the accuracy of estimation, we can also say that there is a trade-off between the estimation error and the estimation lag. Incorporating more PDCs into the active PDC subset leads to more useable measurement data and thus reduced estimation error, but results in large waiting time (for measurement data from local PDCs to arrive) and thus a large lag in oscillation estimation. Whereas incorporating less PDCs in the active PDC subset tends to reduce the lag in estimation but increases estimation error. Numerical simulations in Fig. 19 also confirmed this trade-off.

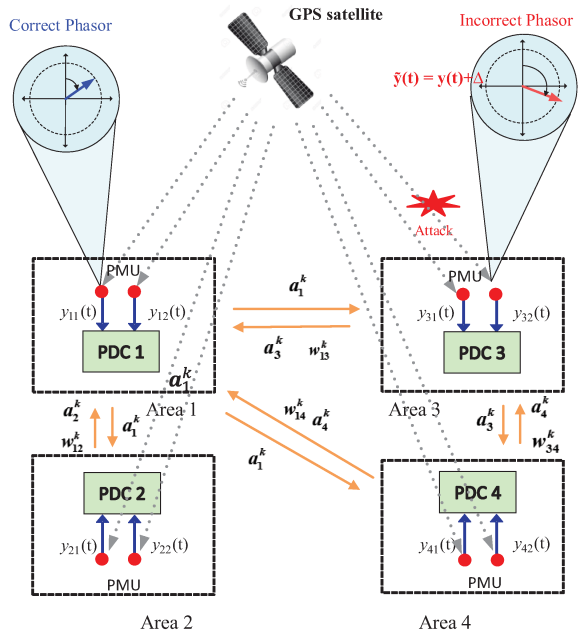

Fig. 2. Decentralized architecture for a 4-area power system network.

V. Decentralized Detection without a central PDC

Algorithms 2 and 3 need a central PDC to update $\boldsymbol{z}$ at each iteration and to broadcast the vector back to the local PDCs. Although this communication architecture preserves the data privacy between all PDCs, it is not resilient to a single point failure of the central PDC under extraneous attacks. To solve this problem, we have to resort to a completely decentralized version of architecture 1 as shown in Fig. 2. In this architecture, each active PDC at each iteration communicates directly with a subset of other active PDCs. Therefore, the need for a central PDC no longer exists. In the absence of a central PDC, the communication patten among PDCs can be described by a communication graph $G = (V, E)$ where each node in $V = \{1, 2, \ldots, N\}$ corresponds to one PDC and E is the edge set with each edge $e_{iv} \in E$ corresponding to a communication link between PDCs i and v. In this paper, we assume that the communication graph G is time-invariant and connected with a path existing between any pair of nodes. We also assume that the communication is synchronous and instantaneous (no latency) among local PDCs. To present the algorithm, we give a graph reformulation of the optimization problem (8):

$$\min_{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N} \sum_{i=1}^{N} \|\hat{\boldsymbol{H}}_i \boldsymbol{a}_i - \hat{\boldsymbol{c}}_i\|^2, \quad \text{s.t.} \quad \boldsymbol{a}_i - \boldsymbol{a}_v = \boldsymbol{0} \text{ for } e_{iv} \in E \tag{13}$$

Since the communication graph is connected, the reformulated problem (13) is equivalent to (8). To facilitate the description of the algorithm, we define the predecessor set P_i and successor set S_i of a PDC i as

$$P_i \triangleq \{v | e_{iv} \in E, v < i\}, \quad S_i \triangleq \{v | e_{iv} \in E, v > i\} \tag{14}$$

and we denote the number of elements in the sets P_i and S_i by n_{P_i} and n_{S_i}, respectively. We now propose a decentralized attack-resilient oscillation monitoring algorithm based on the decentralized ADMM in [46], [4] which can guarantee that the estimated oscillation modes on all local PDCs converge to same values:

Algorithm 4: Attack resilient ADMM without a central PDC

1) Each local PDC i initializes $\boldsymbol{a}_i^0, \boldsymbol{w}_{vi}^0$, for $e_{vi} \in E$.
2) At iteration k, every PDC i $(1 \le i \le N)$:

a) Receives the update of $\boldsymbol{a}_v^{k+1}$ for all $v \in P_i$. For each received $\boldsymbol{a}_v^{k+1}$, PDC i uses the following two rules to detect if it is compromised checks every element of the deviation of $\boldsymbol{a}_v^{k+1}$ from $\boldsymbol{a}_v^k$.

I) ϵ based rule: PDC i checks for every eleme... the absolute difference from the corresponding $\boldsymbol{a}_v^k$ (the estimate from the same local PDC in t... iteration). If for some element, the absolute di... over ϵ, PDC i determines that the v^{th} PDC is co... and its data will be discarded for all iteratio... from the k^{th} iteration.

II) median based rule: PDC i checks for every $\boldsymbol{a}_v^k$ the absolute difference from the correspondi... in the median vector $\tilde{\boldsymbol{a}}_v^k$ of all PDCs in $v \in P_i$ ... transient period (the median is defined on each element as $\tilde{\boldsymbol{a}}_v^k = \text{median}\{\boldsymbol{a}_v^k, v \in P_i\}$, note that here we require that P_i includes at least three PDCs). If for some element, the absolute difference is over δ, then PDC i determines that the v^{th} PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

If the deviation is over ϵ, PDC i determines that the corresponding PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

b) Constructs $\hat{\boldsymbol{H}}_i^k$ and $\hat{\boldsymbol{c}}_i^k$.

c) Updates $\boldsymbol{a}_i$ according to

$$\tilde{\boldsymbol{a}}_i^{k+1} = ((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{H}}_i^k + \rho(n_{p_i} + n_{S_i})\boldsymbol{I})^{-1} \boldsymbol{\kappa}_i^k$$

$$\boldsymbol{\kappa}_i^k = ((\hat{\boldsymbol{H}}_i^k)^T \hat{\boldsymbol{c}}_i^k + \sum_{v \in S_i} \boldsymbol{w}_{iv}^k - \sum_{v \ inP_i} \boldsymbol{w}_{vi}^k + \rho(\sum_{v \in P_i} \boldsymbol{a}_v^{k+1} + \sum_{v \in S_i} \boldsymbol{a}_v^k)).$$

$$\boldsymbol{a}_i^{k+1} = \boldsymbol{a}_i^k + \text{sat}_\epsilon(\tilde{\boldsymbol{a}}_i^{k+1} - \boldsymbol{a}_i^k)$$

Here $\text{sat}_\epsilon(\bullet)$ is the saturation function and is defined as in (10) for a positive constant ϵ.

d) Updates all $\boldsymbol{w}_{vi}$ for $v \in P_i$ as:

$$\boldsymbol{w}_{vi}^{k+1} = \boldsymbol{w}_{vi}^k - \rho(\boldsymbol{a}_v^{k+1} - \boldsymbol{a}_i^{k+1})$$

e) Sends $\boldsymbol{a}_i^{k+1}$ to all PDCs which are either in P_i or in S_i.

f) Sends $\boldsymbol{w}_{vi}^{k+1}$ to $v \in P_i$.

g) Receives $\boldsymbol{a}_v^{k+1}$ and $\boldsymbol{w}_{iv}^{k+1}$ from all $v \in S_i$. For each $\boldsymbol{a}_v^{k+1}$ received from the successors, PDC i checks every element of the deviation of $\boldsymbol{a}_v^{k+1}$ from $\boldsymbol{a}_v^k$. If the deviation is over ϵ, PDC i determines that the corresponding PDC is compromised and its data will be discarded for all iterations onwards from the k^{th} iteration.

In the above algorithm, at each iteration k, the primal variables $\boldsymbol{a}_i^k$ are updated sequentially starting from PDC 1 to PDC

Yongq

replace "checks every

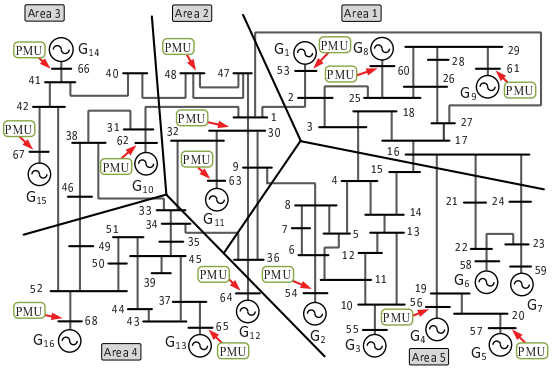

Fig. 3. Distributed architecture for a 5-area power system network.

N using the most recent available vales of $\boldsymbol{a}_v$ for v belonging to its predecessors and successors. PDC i also updates the dual variables $\boldsymbol{w}_{vi}$ for v belonging to P_i. Therefore, if a PDC is compromised by an attack, its successors can immediately detect the attack as the successors are affected right away in the current iteration. However, its predecessors can only detect the attack in the next iteration as the value of a PDC will affect its predecessors only in the next iteration. Furthermore, because in each iteration, the estimation in each PDC is only affected by a subset of all other PDCs, the convergence speed of estimates from different PDCs to the same value is reduced compared with the centralized case, in which the estimation in each PDC is affected by every other PDC (through the central PDC). This is clear from the simulation results (compare Fig. 9 for the centralized case and Fig. 21 for the decentralized case).

Remark 3: It is worth noting that a potential drawback of decentralized schemes is that the required extra communications between local PDCs may increase the communication overhead. The extra communication links could also become targets for a coordinated attack.

VI. Numerical Simulations

We used the IEEE 68-bus system to verify the proposed approach. We divided the entire system into five areas (cf. Fig. 3) with each area having one local PDC and three PMUs. The simulated measurements are obtained using the Power Systems Toolbox [4]. The synchronous generators are assumed to be 6^{th} order for the sake of practicality. A three-phase fault is simulated at the line connecting buses 1 and 2. The measurements are down-sampled, making the sampling period $T = 0.2s$. Our objective is to estimate the post-fault inter-area oscillations. As there are 16 generators, our algorithm should ideally solve a 96^{th} order polynomial. However, many of these 96 modes are negligible and it was shown in [4] that 40 modes suffice to capture the inter-area oscillations. We used $\rho = 10^{-9}$ in the simulation. The proposed algorithms are lightweight in computation. In fact, in our Matlab simulations on a computer with Intel 1.7 GHz CPU and 8 GB memory, it took approximately 2.56 milliseconds to run our algorithm for 50 iterations.

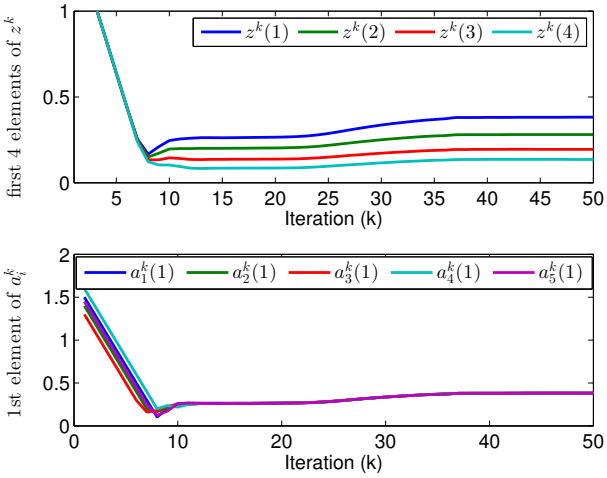

Fig. 4. $\boldsymbol{z}^k$ in the central PDC and $\boldsymbol{a}_i^k$ in local PDCs in Algorithm 2.

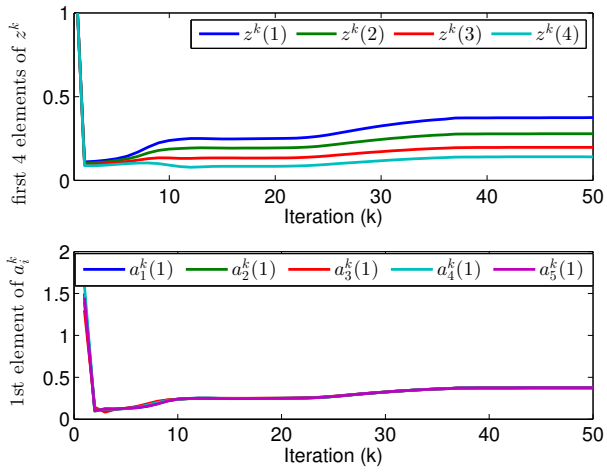

Fig. 5. $\boldsymbol{z}^k$ in the central PDC and $\boldsymbol{a}_i^k$ in local PDCs in Algorithm 1.

A. Simulation results in the presence of a central PDC and synchronous communications

We first checked the convergence property of the proposed algorithm 2 in the absence of attacks. We set ϵ to 0.2 to run the simulations. Fig. 4 shows the evolution of the first 4 elements of $\boldsymbol{z}^k$ on the central PDC and the first element of $\boldsymbol{a}^k$ on the five local PDCs. It can be seen that they converge to the same steady-state values as in the original ADMM algorithm (cf. Fig. 5). The selected estimated modes σ and Ω in (3) (the solid lines in Fig. 6) also converge to the actual values obtained from the Power System Toolbox (represented by dashed lines in Fig. 6). In the simulations, the absolute time for each iteration was on the sub-millisecond level. The total running time for all 50 iterations was around 16 milliseconds.

We then evaluated the performance of the proposed algorithm in the presence of an attack on PDC 1. Following [3] showing that GPS spoofing leads to a constant drift, we assume that a constant drift $\Delta = 0.2$ occurred on PDC 1's estimate of $\boldsymbol{a}_i^k$ at iteration no. 26. The estimated modes with the original algorithm are given in Fig. 8. It can be seen that the attack leads to erroneous estimation. Whereas with the

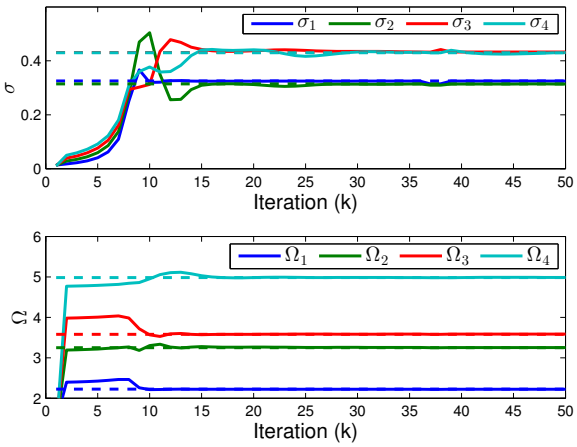

Fig. 6. Four estimated oscillation modes using the proposed Algorithm 2.

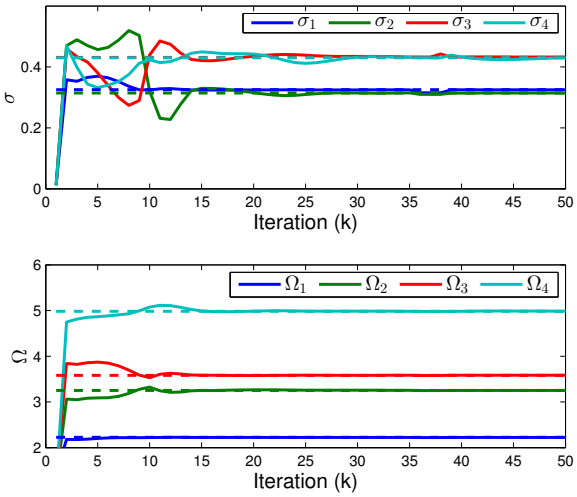

Fig. 7. Four estimated oscillation modes using the original Algorithm 1.

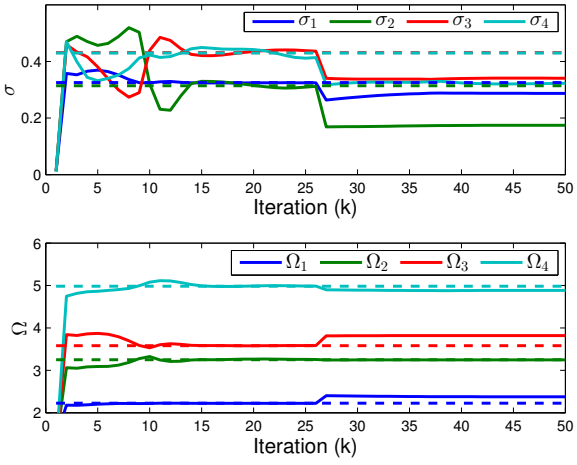

Fig. 8. Estimated modes in Algorithm 1 in the presence of an attack.

proposed algorithm, the attacked PDC was quickly detected and isolated, which guarantees a correct estimation (cf. Fig. 9).

We also simulated the influence of ϵ on the estimation error

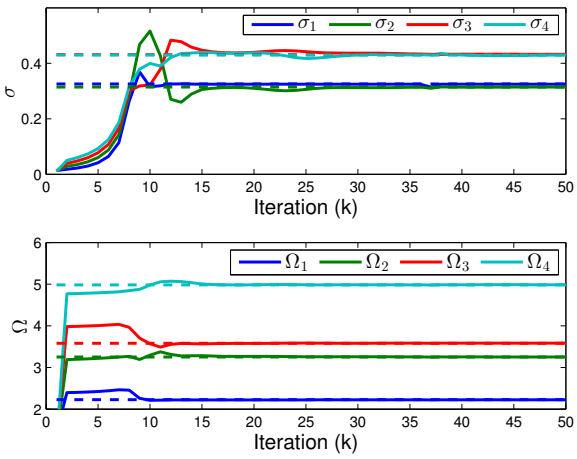

Fig. 9. Estimated modes in Algorithm 2 in the presence of an attack.

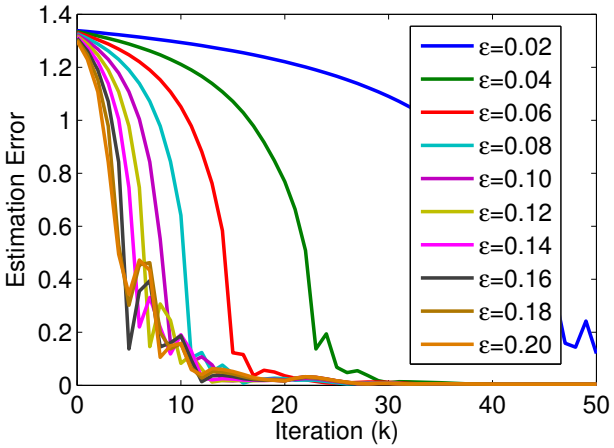

Fig. 10. The influence of the magnitude of ϵ in Algorithm 2 on the final estimation error of σ.

of σ. The results are given in Fig. 10. It can be seen that the proposed algorithm 2 is insensitive to the magnitude of ϵ in correctly estimating the oscillation modes, although it's convergence speed decreases with a decrease in ϵ.

To test the effectiveness of the algorithm in the presence of multiple attacks, we also ran Algorithm 2 when both PDC 1 and PDC 5 were compromised with a constant offset value 0.2. In the simulation, the offset on the outputs of PDC 1 and PDC 5 was initiated at iterations 25 and 35, respectively. Algorithm 2 successfully detected and isolated both compromised PDCs, and correctly estimated the oscillation modes, as shown in Fig. 11.

We also evaluated the effectiveness of the "median based rule" in detecting attackers that can evade the "ϵ based rule" by using very small deviations. According to simulation results, we found that the transient period is less than 10 iterations. So we activated the "median based rule" after 10 iterations with δ set to 0.01. To emulate attacks with small deviations, we induced a 0.005-per-iteration offset to the first element of $\boldsymbol{a}_1^k$ starting from iteration number 26. The evolution of $\boldsymbol{a}_1^k$ is shown in Fig. 12. The detection mechanism successfully captured this attack at iteration number 29, and hence guaranteed the correct estimation of the oscillation modes, as shown in

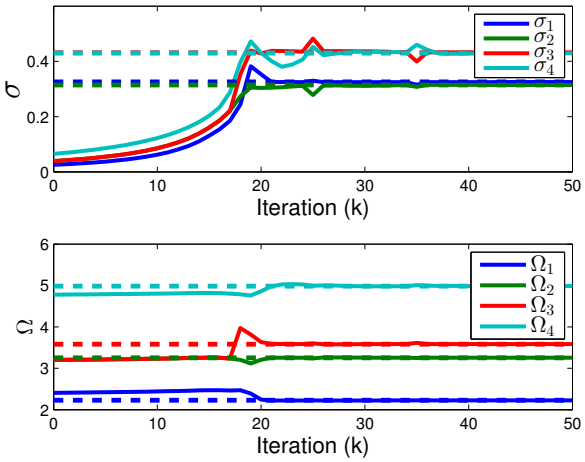

Fig. 11. Estimated modes in Algorithm 2 in the presence of two attacks on PDC 1 and PDC 5, respectively.

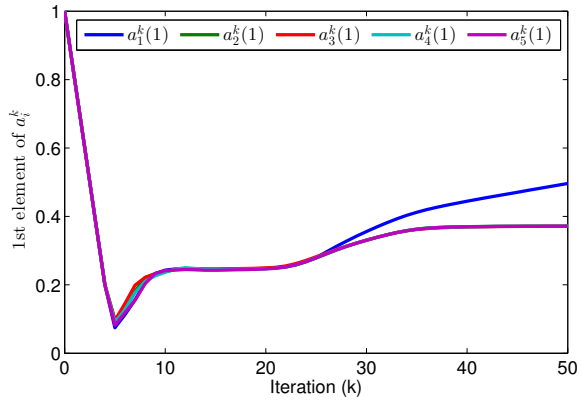

Fig. 12. $\boldsymbol{a}_i^k$ in local PDCs with $\boldsymbol{a}_1^k$ subject to attack-induced small deviations (0.005/iteration) starting from iteration number 26.

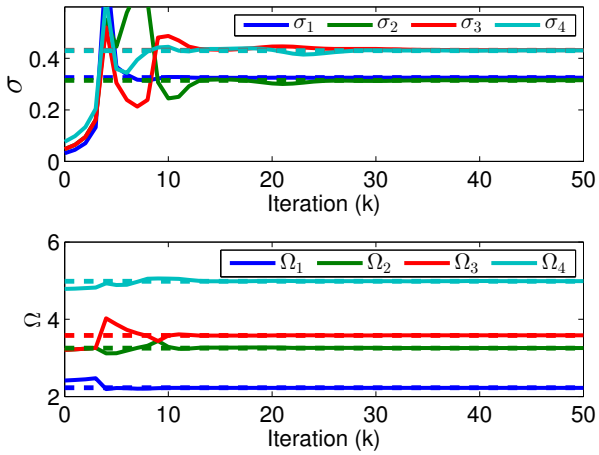

Fig. 13. Four estimated oscillation modes using the proposed Algorithm 2 with $\boldsymbol{a}_1^k$ subject to attack-induced small deviations (0.005/iteration) starting from iteration number 26.

Fig. 13.

The "median based rule" was also confirmed effective in capturing attacks holding some $\boldsymbol{a}_i^k$ to a constant value. More specifically, in the simulation, we assumed that the first element of $\boldsymbol{a}_1^k$ was attacked and stopped updating starting from

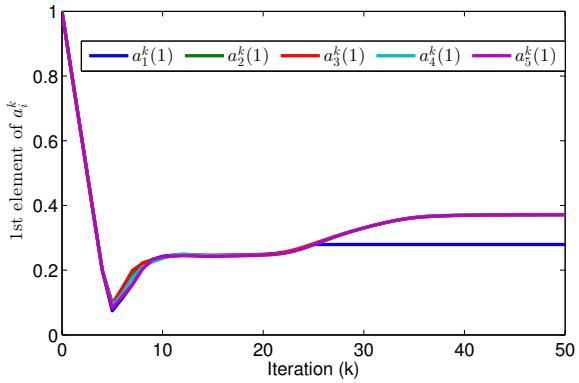

Fig. 14. $\boldsymbol{a}_i^k$ in local PDCs with $\boldsymbol{a}_1^k$ subject to an attack holding it to a constant value starting from iteration number 26.

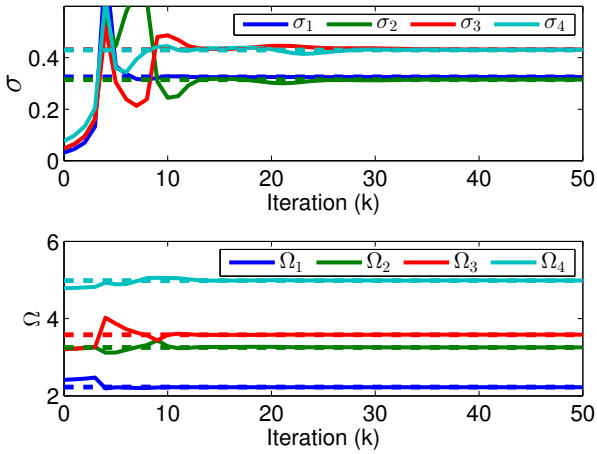

Fig. 15. Four estimated oscillation modes using the proposed Algorithm 2 with $\boldsymbol{a}_1^k$ subject to an attack holding it to a constant value starting from iteration number 26.

iteration number 26, i.e., the first element of $\boldsymbol{a}_1^k$ was set to the first element of $\boldsymbol{a}_1^{26}$ for all $k = 27, 28, \ldots$ (cf. Fig. 14). The "median based rule" successfully detected this attack and guaranteed a correct estimation, as illustrated in Fig. 15.

Besides GPS spoofing, GPS signal loss can also affect phasor angle measurements [48] and thus affect oscillation mode estimation. Therefore, we also ran the "median based rule" under phasor angle drifting rate 0.0002 rad/s obtained from real power grid measurements [48]. According to the relationship in (6), we obtained numerically that such a drifting rate could induce 0.0002-per-iteration offset to the first element of $\boldsymbol{a}_1^k$. The drift was set to start at iteration number 26, and was detected at iteration number 97 (cf. the evolution of $\boldsymbol{a}_1^k$ in Fig. 16). The estimated oscillation modes are given in Fig. 17, which confirmed the resilience of the approach to GPS signal loss.

B. Simulation results in the presence of a central PDC and asynchronous communications

We also simulated Algorithm 3 in the presence of a central PDC and asynchronous communications. To emulate the asynchronous communication patterns, we chose active PDCs randomly with each PDC having a equal probability of 0.5 for being active. In the absence of attacks, the algorithm correctly

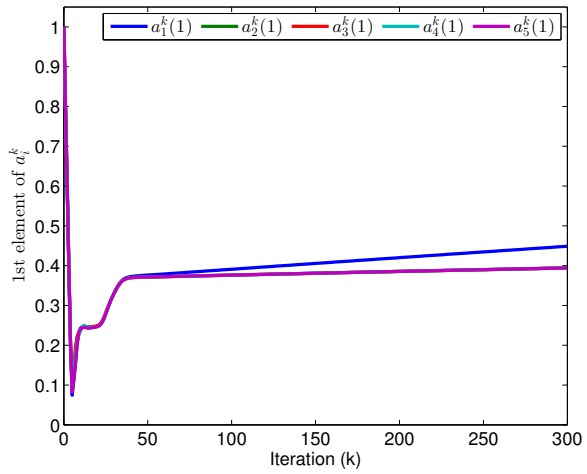

Fig. 16. $\boldsymbol{a}_i^k$ in local PDCs with $\boldsymbol{a}_1^k$ subject to GPS-signal-loss-induced small deviations (0.0002/iteration) starting from iteration number 26.

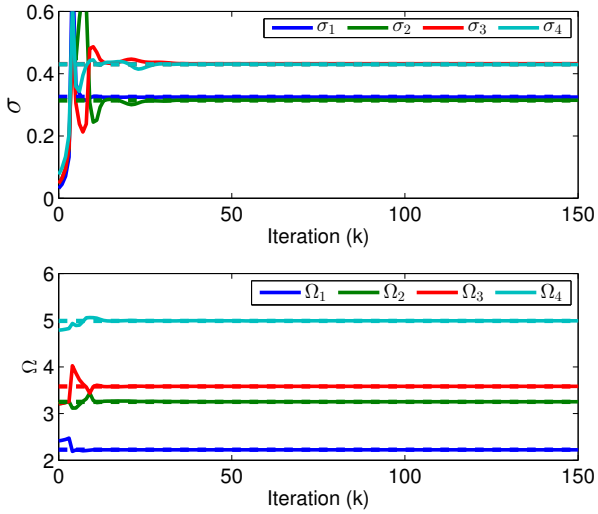

Fig. 17. Four estimated oscillation modes using the proposed Algorithm 2 with $\boldsymbol{a}_1^k$ subject to GPS-signal-loss-induced small deviations (0.0002/iteration) starting from iteration number 26.

estimated the oscillation modes. In the presence of an attack on PDC 1, the algorithm quickly detected the attack and isolated PDC 1, and hence could still successfully estimate the oscillation modes (cf. Fig. 18).

We also numerically verified the trade-off between estimation error and estimation lag for Algorithm 3. In the simulation the communication latencies were uniformly distributed between 0 and 7 iterations. We defined a local PDC to be within the active subset when the latency on its communication link to the central PDC (which is time varying) is no larger than i iterations ($i = 1, 2, \cdots, 7$), leading to a lag (delay) of i iterations in the oscillation estimation. The total estimation error of the four most significant oscillation modes under different lags i at iteration number 50 is plotted in Fig. 19. It is evident that there is a trade-off between estimation error and estimation lag.

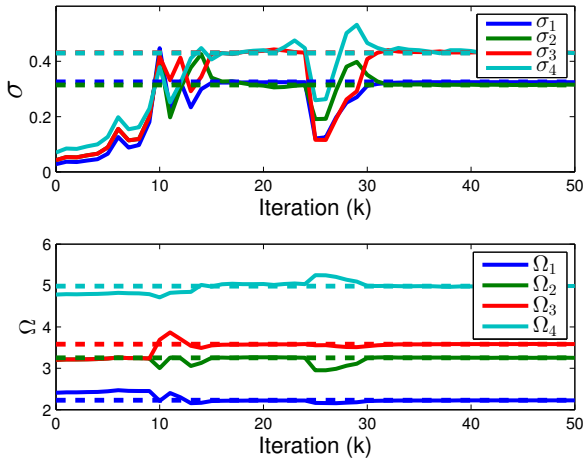

Fig. 18. Estimated oscillation modes in Algorithm 3 in the presence of an attack.

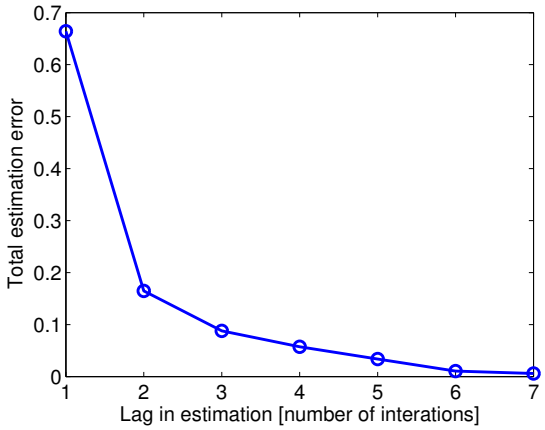

Fig. 19. Total estimation error in four most significant oscillation modes under different lags of estimation.

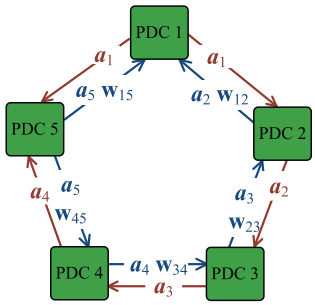

Fig. 20. Decentralized communication pattern.

C. Simulation results in the absence of a central PDC

We also simulated Algorithm 4 in the absence of a central PDC. The communication architecture is given in Fig. 20. In the absence of attacks, the algorithm correctly estimated the oscillation modes. In the presence of an attack on PDC 5, the algorithm quickly detected the attack and isolated PDC 5, and hence could still successfully estimate the oscillation modes (cf. Fig. 21).

VII. Conclusion

We proposed a real-time distributed wide-area oscillation estimation approach that is resilient to GPS spoofing attacks.

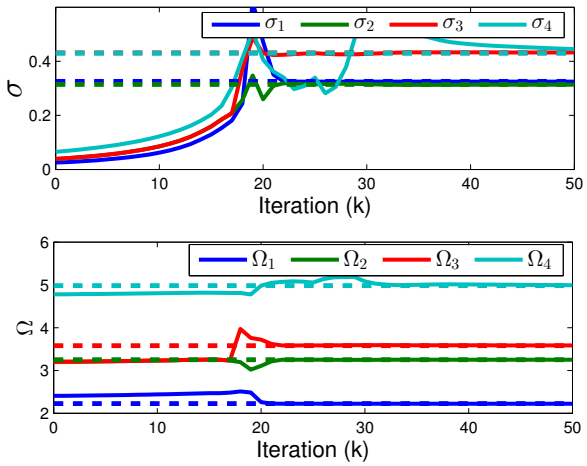

Fig. 21. Estimated modes in Algorithm 4 in the presence of an attack on PDC 5.

The approach can be used in the presence of asynchronous communications due to heterogenous computational speeds and heterogenous communication latencies. It can also be used in a completely decentralized scenario without a central PDC. In fact, the approach is essentially based on designed distributed agreement and could potentially be adapted as solutions to other types of attacks on distributed optimization and control of a power system. For example, the fault-current injection attacks in [39] affect multiple PMU measurements and can likely be tackled under the proposed detection framework. Numerical simulations confirmed that the proposed algorithms can detect small attacks and guarantee correct estimation in the presence of attacks.

REFERENCES

[1] A. G. Phadke and J. S. Thop. *Synchronized Phasor Measurements and Their Applications*. Springer, New York, 2008.

[2] S. Toscani, C. Muscas, and P. A. Pegoraro. Design and performance prediction of space vector-based PMU algorithms. *IEEE Transactions on Instrumentation and Measurement*, 66:394–404, 2017.

[3] D. P. Shepard, T. E. Humphreys, and A. A. Fansler. Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks. *International Journal of Critical Infrastructure Protection*, 5:146–153, 2012.

[4] S. Nabavi, J. Zhang, and A. Chakrabortty. Distributed optimization algorithms for wide-area oscillation monitoring in power systems using inter-regional PMU-PDC architectures. *IEEE Trans. Smart Grid*, 6:2529–2538, 2015.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating directional method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[6] N. Zhou, D. Trudnowski, J. Pierre, and W. Mittelstadt. Electromechanical mode online estimation using regularized robust rls methods. *IEEE Transactions on Power Systems*, 23:1670–1680, 2008.

[7] A. R. Messina and V. Vittal. Nonlinear, non-stationary analysis of interarea oscillations via hilbert spectral analysis. *IEEE Transactions on Power Systems*, 21:1234–1241, 2006.

[8] L. Heng, D. B. Work, and G. Gao. GPS signal authentication from cooperative peers. *IEEE Transactions on Intelligent Transportation Systems*, 16:1794–1805, 2015.

[9] X. Jiang, J. Zhang, B. Harding, J. J. Makela, and A. D. Dominguez-Garcia. Spoofing GPS receiver clock offset of phasor measurement units. *IEEE Transactions on Power Systems*, 28:3253–3262, 2013.

[10] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li. Time synchronization attack in smart grid: impact and analysis. *IEEE Transactions on Smart Grid*, 4:87–98, 2013.

[11] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys. GPS spoofing detection via dual-receiver correlation of military signals. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4):2250–2267, 2013.

[12] L. Heng, D. B. Work, and G. X. Gao. GPS signal authentication from cooperative peers. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1794–1805, 2015.

[13] R. T. Ioannides, T. Pany, and G. Gibbons. Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques. *Proceedings of the IEEE*, 104(6):1174–1194, 2016.

[14] D. Borio and C. Gioia. A sum-of-squares approach to gnss spoofing detection. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1756–1768, 2016.

[15] Y. Fan, Z. Zhang, M. Trinkle, A. D. Dimitrovski, J. Song, and H. Li. A cross-layer defense mechanism against GPS spoofing attacks on pmus in smart grids. *IEEE Transactions on Smart Grid*, 6:2659–2668, 2015.

[16] D. Yu, A. Ranganathan, T. Locher, S. Capkun, and D. Basin. Short paper: detection of GPS spoofing attacks in power grids. In *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*, pages 99–104. ACM, 2014.

[17] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problems. *ACM Transactions on Programming Languages and Systems*, 4:382–401, 1982.

[18] Y. Q. Wang, J. Hespanha, and A. Chakrabortty. Distributed monitoring of wide-area oscillations in the presence of GPS spoofing attacks. In *IEEE PES General Meeting*, Boston, 2016.

[19] P. A. Pegoraro, A. Meloni, L. Atzori, P. Castello, and S. Sulis. PMU-based distribution system state estimation with adaptive accuracy exploiting local decision metrics and iot paradigm. *IEEE Transactions on Instrumentation and Measurement*, 66:704–714, 2017.

[20] C. Muscas, M. Pau, P. A. Pegoraro, and S. Sulis. Uncertainty of voltage profile in PMU-based distribution system state estimation. *IEEE Transactions on Instrumentation and Measurement*, 65:988–998, 2016.

[21] A. Sharma, S. C. Srivastava, and S. Chakrabarti. A cubature kalman filter based power system dynamic state estimator. *IEEE Transactions on Instrumentation and Measurement*, 66:2036–2045, 2017.

[22] M. Pau, F. Ponci, A. Monti, S. Sulis, C. Muscas, and P. A. Pegoraro. An efficient and accurate solution for distribution system state estimation with multiarea architecture. *IEEE Transactions on Instrumentation and Measurement*, 66:910–919, 2017.

[23] A. Angioni, J. Shang, F. Ponci, and A. Monti. Real-time monitoring of distribution system based on state estimation. *IEEE Transactions on Instrumentation and Measurement*, 65:2234–2243, 2016.

[24] S. Sarri, L. Zanni, M. Popovic, J. L. Boudec, and M. Paolone. Performance assessment of linear state estimators using synchrophasor measurements. *IEEE Transactions on Instrumentation and Measurement*, 65:535–548, 2016.

[25] S. Kar and G. Hug. Distributed robust economic dispatch in power systems: a consensus + innovations approach. In *IEEE PES General Meeting*, 2012.

[26] Z. Zhang and M. Y. Chow. Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid. *IEEE Transactions on Power Systems*, 27:1761–1768, 2012.

[27] E. DallAnese, H. Zhu, and G. B. Giannakis. Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid*, 4:1464–1475, 2013.

[28] A. D. Dominguez-Garcia and C. N. Hadjicostis. Coordination of distributed energy resources for provision of ancillary services: architectures and algorithms. *Encyclopedia of Systems and Control*, 2014.

[29] T. Erseghe. Distributed optimal power flow using ADMM. *IEEE Transactions on Power Systems*, 29:2370–2380, 2014.

[30] J. Chen and A. Abur. Placement of PMUs to enable bad data detection in state estimation. *IEEE Transactions on Power Systems*, 21(4):1608–1615, 2006.

[31] B. Gou and R. G. Kavasseri. Unified PMU placement for observability and bad data detection in state estimation. *IEEE Transactions on Power Systems*, 29(6):2573–2580, 2014.

[32] M. B. Do Coutto Filho, J. C. S. de Souza, and M. A. R. Guimaraens. Enhanced bad data processing by phasor-aided state estimation. *IEEE Transactions on Power Systems*, 29(5):2200–2209, 2014.

[33] N. G. Bretas, A. S. Bretas, and A. C. P. Martins. Convergence property of the measurement gross error correction in power system state estimation,

using geometrical background. *IEEE Transactions on Power Systems*, 28(4):3729–3736, 2013.

[34] J. Zhao, G. Zhang, M. La Scala, and Z. Wang. Enhanced robustness of state estimator to bad data processing through multi-innovation analysis. *IEEE Transactions on Industrial Informatics*, 13(4):1610–1619, 2017.

[35] T. T. Kim and H. V. Poor. Strategic protection against data injection attacks on power grids. *IEEE Transactions on Smart Grid*, 2(2):326–333, 2011.

[36] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, 14(1):13, 2011.

[37] J. Kim and L. Tong. On phasor measurement unit placement against state and topology attacks. In *Proceedings of IEEE International Conference on Smart Grid Communications*, pages 396–401. IEEE, 2013.

[38] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 2017.

[39] A. S. Dobakhshari and A. M. Ranjbar. A wide-area scheme for power system fault location incorporating bad data detection. *IEEE Transactions on Power Delivery*, 30(2):800–808, 2015.

[40] S. B. Andrade, M. Pignati, G. Dan, M. Paolone, and J. Le Boudec. Undetectable PMU timing-attack on linear state-estimation by using rank-1 approximation. *IEEE Transactions on Smart Grid*, 2016.

[41] P. Risbud, N. Gatsis, and A. Taha. Assessing power system state estimation accuracy with GPS-spoofed PMU measurements. In *Proceedings of the 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference*, pages 1–5. IEEE, 2016.

[42] P. Kundur. *Power system stability and control*. McGraw-hill, New York, 1994.

[43] J. F. Hauer, C. J. Demeure, and L. L. Scharf. Initial results in prony analysis of power system response signals. *IEEE Transactions on Power Systems*, 7:1559–1564, 1992.

[44] IEEE standard for synchrophasor measurements for power systems. *IEEE Standard C37.118.1-2011*, 2011.

[45] IEEE power and energy society (PES) technical report. *IEEE Transactions on Power Systems*, available online at: http://sites.ieee.org/pes-resource-center/files/2013/11/TR15_Modal_Ident _TF_Report.pdf.

[46] E. Wei and A. Ozdaglar. On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers. In *Proceedings of IEEE Global Conference on Signal and Information Processing*, 2013.

[47] R. Zhang and J. T. Kwok. Asynchronous distributed ADMM for consensus optimization. In *Proceedings of the 31st International Conference on Machine Learning*, page 17011709, Beijing, China, 2014.

[48] W. X. Yao, Y. Liu, D. Zhou, Z. H. Pan, J. C. Zhao, M. Till, L. Zhu, L. W. Zhan, Q. Tang, and Y. L. Liu. Impact of GPS signal loss and its mitigation in power system synchronized measurement devices. *IEEE Transactions on Smart Grid*, 2016.