State estimation in power distribution networks with poorly synchronized measurements

Saverio Bolognani, Ruggero Carli and Marco Todescato

***Abstract*—We consider the problem of designing a state estimation architecture for the power distribution grid, capable of collecting raw measurements from low-end phasor measurement units, processing the data in order to minimize the effects of measurement noise, and presenting the state estimate to the control and monitoring applications that need it. The proposed approach is leader-less, scalable, and consists in the distributed solution of a least square problem which takes into explicit consideration the inexact time synchronization of inexpensive PMUs. Two algorithms are proposed: the first one is a specialization of the alternating directions method of multipliers (ADMM) for which a scalable implementation is proposed; the second one is an extremely lightweight Jacobi-like algorithm. Both algorithms can be implemented locally by local data aggregators, or by the individual PMUs, in a peer-to-peer fashion. The proposed approach is validated via simulations on the IEEE 123 test feeder, considering the IEEE standard C37.118-2005 for PMUs. We also analyze the performance of a distributed control algorithm that makes use of the grid state estimation and that we adopted as a prototype. We show that, via the proposed state estimation architecture, feedback control of the distribution grid becomes viable also with poorly synchronized sensors, and possibly also without GPS-based synchronization modules.**

I. Introduction

The electric grid is currently undergoing a deep renovation process towards the so-called *smart grid*, featuring larger hosting capacity, widespread penetration of renewable energy sources, better quality of the service, and higher reliability. One of the major aspects of this modernization is the widespread deployment of dispersed measurement, monitoring, and actuation devices.

In this paper, we consider one specific thrust, which is the deployment of phasorial measurement units (PMUs) in the medium and low voltage power distribution grid. The presence of PMUs is quite uncommon in today's power distribution networks. However, this scenario has become the subject of recent research efforts in the power systems community, including a 3-year research project involving University of California together with the Power Standards Lab and Lawrence Berkeley National Lab [1]. Previous investigation on the subject includes the experimental deployment and the extensive measurement campaign in [2], and the analysis proposed in [3], [4], [5] for the possible applications of such measurement devices, also called μPMUs, in the power distribution grid.

There is a substantial consensus on the many opportunities offered by the deployment of a live network of μPMUs, including diagnostic routines (e.g. for unplanned islanding and fault detection), power flow estimation on the power lines (by knowing the phasorial nodal voltage differences across the entire grid), and feedback control strategies for coordination of renewable resources, islanding, and volt/VAR compensation.

There are a number of unresolved issues that need to be addressed, in order to make these applications possible. As the cost of PMUs depends on their sensing accuracy, noisy and low-end device will necessarily be preferred in order to keep the cost of large scale deployment acceptable. In particular, time synchronization between different PMUs is a major technological issue, which is generally tackled via GPS modules that can provide timestamping of the data.

Following the approach that is typically adopted in wireless sensor networks [6], [7], in this paper we propose a solution to these issues that exploits the large number of sensors and their communication capabilities to compensate for their modest performance. We propose a leader-less and distributed state estimation architecture, in which PMUs communicate with their neighbor peers in order to improve the quality of their measurements. Such improved measurements are then made available locally to the monitoring and control algorithms that may need them, in a transparent way and without any aggregation of the data at a central server (see Figure 1).

We design two different state estimation algorithms that can be adopted in order to implement the proposed architecture. Both have pros and cons, that we discuss. Compared to the existing approaches to state estimation for power transmission networks (which are reviewed later in this section), we focus on some specific features that are of crucial importance in the scenario of medium voltage power distribution grids:

- we need to deal effectively with poorly synchronized measurements;
- we cannot assume large X/R impedance ratio of the power lines;
- we only allow measurements that can be performed by devices connected to the grid buses, and no power flow and current measurements on the power lines;
- we do not assume redundancy of measurement locations (instead, we exploit the fact of being able to sense both voltage and current injection at each bus);
- the computational effort must remain limited as the grid grows in size;
- we aim at leader-less algorithms (in which no grid supervisor or central data aggregator is present);
- we allow a peer-to-peer architecture, in which each PMU

S. Bolognani is with the Laboratory on Information and Decision Systems, Massachusetts Institute of Technology, Cambridge (MA), USA email: saverio@mit.edu.

R. Carli and M. Todescato are with Department of Information Engineering, University of Padova, Padova (PD), Italy email {carlirug | todescat}@dei.unipd.it

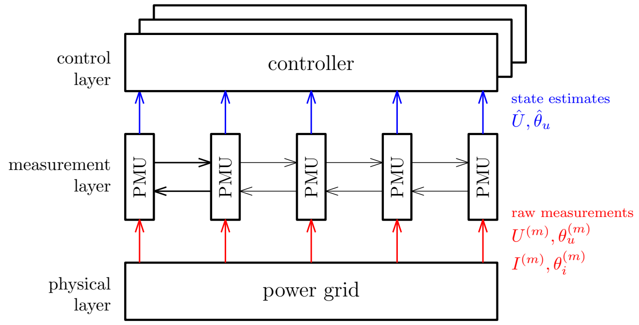

Fig. 1. The panel shows a functional representation of the proposed architecture. Each bus is provided with a PMU, which measures bus voltage and current. These device communicate with their neighbors, in order to process the raw data and obtain a better estimate of the system state. They constitute the middle *measurement layer* that we are considering in this paper. The state estimates computed in this layer are then transmitted to the control layer, where different monitoring and optimization algorithm can coexist.

is provided with its own computational capabilities, and can communicate with its neighbor PMUs (even if we also consider the possibility of local data aggregators in the algorithm, as it might be a convenient architecture when combined with some hierarchical communication infrastructures [8]).

The paper is organized as follows. In the remainder of this section, we review the main related technical literature. In Section II we introduce a model for the power grid and for the measurement errors, where time synchronization errors are explicitly considered. Based on this model, we detail the least-square problem that has to be solved in order to find the maximum-likelihood estimation of the grid state (Section III). In Section IV we illustrate how the problem can be decomposed into sub-problems, each one corresponding to a subset of the PMUs (possibly to individual PMUs, in the peer-to-peer case). In Section V we propose two different algorithmic approaches. The first approach is based on a distributed implementation of the Alternating Direction Method of Multipliers (ADMM, [9]). In particular we tailored the algorithm introduced in [10] to the case of quadratic cost functions showing that it reduces to a linear iterative algorithm. The second approach is a leader-less distributed Jacobi-like algorithm, where each agent has to solve an extremely simple optimization problem, for which a closed form solution is provided. Of note is the fact that both algorithms can be implemented in a scalable way, in which every agent is only required to store a portion of the entire state of the system. In Section VI we show how both these algorithms are effective in improving the quality of the measurements, providing a consistent and accurate estimate of the node voltages, and we evaluate their performance both in the multi-area approach and in the special case of a peer-to-peer architecture. We then consider a specific application belonging to the class of problems that can be addressed by the deployment of a live network of μPMUs. Such application is the coordinated control of the power inverters of microgenerators to provide volt/VAR compensation, i.e. reactive power compensation for voltage support and power loss minimization [11], [12], [13], [14]. We consider the strategy proposed in [15], and we show that the measurement errors in standard low-end PMUs are unacceptable for such task. When the proposed state estimation algorithms are employed, instead, the improved measurements become accurate enough to be used effectively. The resulting state estimation is particularly robust against synchronization errors, to a degree that it is possible to envision implementations in which PMUs are not provided with GPS modules, relying instead on standard synchronization routines over the communication network.

A. Related work

Several solutions alternative to the centralized computation of the state of a electric grid have been proposed in the literature. The most common approach is based in a two level architecture requiring a global coordinator [16], [17], [18], [19], [20], [21]. In this multi-area approach the grid is split into macro-areas, each of them equipped with a data processor with communication and computational capabilities. Each processor estimates the state of the sub-area which has been assigned to it, excluding boundary bus measurements, and sends this estimate to the global coordinator. The global coordinator combines all these estimates making them compatible to the boundary bus measurements obtained at a interconnection level. Of note is that this approach becomes uninteresting in the case in which each macro area reduces to one single PMU, because the global coordinator would have to face the very same problem of a centralized state estimator.

Leader-less solutions include approximate algorithms developed from the optimality conditions involved [22], [23]. Beyond requiring local observability, these algorithm are not always guaranteed to converge. The auxiliary principle is invoked in [24], with the drawback that several paramenters have to be tuned. Also the distributed algorithms proposed in [25], [26], adopt the multi-area approach. Remarkably, the algorithm in [25] is shown to be provably convergent in finite time to the optimal solution of a classical weighted least problem. However each area is envisioned to maintain a copy of the entire high-dimensional state vector, drastically increasing the communication and computation effort required at each iteration. In [26] the overall system is decomposed into subsystems, each of them provided with a slack bus monitored by a PMU. Each subsystem obtain its own local estimate, and these estimates are then coordinated based on the PMU measurements, re-estimating the boundary state variables at the aggregation level. However, the global solution obtained through this two-steps procedure is not guaranteed to coincide with the optimal centralized solution, and no analysis of potential synchronization errors is included.

To address the conventional least-squares power system state estimation problem, the authors in [10] propose a novel algorithm which is a local version of the classical ADMM. When dealing with the exact non-linear model of the electric grid, the functional costs to be minimized are non-convex and, in general, the behavior of the ADMM is characterized by slow transient performance. For this reason, typically, the exact models are iteratively linearized via the Gauss-Newton method, or by resorting to the so-called DC approximation

[27]. Either way, one arrives at classical linear models which are just first-order approximations of the true model. Even though the ADMM-based algorithm introduced in [10] exhibits good performance in simulations, a complete proof of its convergence is not provided. We stress the fact that one of the two approaches that we propose in this paper is an extension of the local ADMM algorithm of [10] to the minimization of quadratic functions. These quadratic functions are obtained by casting the state estimation problem as a classical least-squares problem and by adopting a specific linearized model which also allows to deal with the synchronization errors in the measurements provided by the PMUs. The ADMM algorithm we propose in this paper reduces to linear iterations and it is shown to be provably convergent. Due to space limitations, we do not report here the proof of its convergence and we refer the interested reader to the document [28], where we provide all the technical details.

II. Power distribution grid model

In this section, we introduce some useful notation and we provide a description and a mathematical model for the power distribution grid.

A. Power grid model

We model a power distribution grid as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which edges represent the power lines, and the n nodes ($n = |\mathcal{V}|$) represent buses and also the point of common coupling (PCC), i.e., the point of connection of the power distribution grid to the trasmission grid. The grid topology in represented via the incidence matrix A, defined as

$$[A]_{\ell v} = \begin{cases} \text{-1} & \text{if } \ell \text{ starts from node } v \\ 1 & \text{if } \ell \text{ ends in node } v \\ 0 & \text{otherwise.} \end{cases}$$

We consider a balanced three-phase grid, in which all steady state currents and voltages can therefore be represented as a complex number $y = Y \exp(j\theta_y)$ of absolute value Y and phase θ_y (measured with respect to an arbitrary common time reference). Exploiting this notation, the steady state of the network is described by the following system variables:

- $u \in \mathbb{C}^n$, where u_v is the voltage at bus v;
- $i \in \mathbb{C}^n$, where i_v is the current injected at bus v.

Let $Z = \operatorname{diag}(z_\ell, \ell \in \mathcal{E})$ be the diagonal matrix of line impedances, in which z_ℓ is the impedance of the power line corresponding to edge ℓ. From Kirchhoff's laws we have

$$i = Lu, \tag{1}$$

where L is the complex-valued weighted Laplacian matrix (also known as the grid admittance matrix)

$$L = A^T Z^{-1} A.$$

In the following, when we refer to the power grid state, we refer to the vector of bus voltages u, which describe the system working point uniquely.

B. PMU model

Every bus of the power distribution grid is provided with a PMU, which measures the bus voltage $u_v = U_v \exp(j\theta_{u_v})$ and the injected current $i_v = I_v \exp(j\theta_{i_v})$. Compared to other measurement setups presented in the power system literature, especially for the high voltage power transmission network, the setup presented here is therefore characterized by the fact that only nodal measurements are available, while power lines are unmonitored.

We assume that the different PMUs sense the grid at synchronous, evenly distributed, times $t \in \{kT, k \in \mathbb{N}\}$, where $T > 0$ is a predetermined sample time, as dictated also by the IEEE standard C37.118-2005 [29].

At every measurement time t, the PMUs at every bus v obtain the following noisy data

$$\begin{aligned} U_v^{(m)} &= U_v + e_{U_v} \\ \theta_{u_v}^{(m)} &= \theta_{u_v} + e_{\theta_{u_v}} + e_{\text{sync},v} \\ I_v^{(m)} &= I_v + e_{I_v} \\ \theta_{i_v}^{(m)} &= \theta_{i_v} + e_{\theta_{i_v}} + e_{\text{sync},v} \end{aligned} \tag{2}$$

where the different measurement error terms are caused by independent physical causes and can therefore be assumed uncorrelated:

- The magnitude error terms e_{U_v} and e_{I_v} are caused by quantization of the acquisition devices and by harmonic distortion of the signals.
- The phase error terms $e_{\theta_{u_v}}$ and $e_{\theta_{i_v}}$ depend on the sampling time of the acquisition devices and on the algorithm employed for the estimation of the voltage-current phase difference, traditionally denoted by ϕ.
- The synchronization error $e_{\text{sync},v}$ represents the phase error due to inexact time synchronization between different PMUs, and is the same for both the voltage measurement and the current measurement at the same PMU.

We assume that PMUs are homogeneous (even if the proposed formulation could easily account for heterogeneity of the measurement devices) and thus their errors exhibit the same probability distribution. We adopt a Gaussian distribution for all these error terms, and therefore we have for all $v \in V$

$$\begin{aligned} e_{U_v} &\sim \mathcal{N}(0, \sigma_U^2) \\ e_{I_v} &\sim \mathcal{N}(0, \sigma_I^2) \\ e_{\theta_{u_v}}, e_{\theta_{i_v}} &\sim \mathcal{N}(0, \sigma_\theta^2) \\ e_{\text{sync},v} &\sim \mathcal{N}(0, \sigma_{\text{sync}}^2). \end{aligned} \tag{3}$$

A discussion about the variance of these error terms is postponed to Section VI, where typical values given by industrial standards, are reported. Notice that, because we are modeling a time synchronization error as a phase angle, a wrapped normal distribution [30] (or the approximated Fisher-von Mises distribution) should be adopted for that specific term. The difference is however absolutely negligible, because of the extremely small size of typical phase measurement errors.

Let us stack all the measurement errors in the vectors

$e_U, e_{\theta_u}, e_I, e_{\theta_i}, e_{\text{sync}}$, and let

$$e = \begin{bmatrix} e_U \\ e_{\theta_u} + e_{\text{sync}} \\ e_I \\ e_{\theta_i} + e_{\text{sync}} \end{bmatrix}$$

be the cumulative vector of all noise terms, in the order introduced in (2). Then the correlation matrix R for the noise is

$$\begin{aligned} R_e &= \mathbb{E}[ee^T] \\ &= \begin{bmatrix} \sigma_U^2 \mathbf{I}_n & & & \\ & (\sigma_\theta^2 + \sigma_{\text{sync}}^2)\mathbf{I}_n & & \sigma_{\text{sync}}^2 \mathbf{I}_n \\ & & \sigma_I^2 \mathbf{I}_n & \\ & \sigma_{\text{sync}}^2 \mathbf{I}_n & & (\sigma_\theta^2 + \sigma_{\text{sync}}^2)\mathbf{I}_n \end{bmatrix}. \end{aligned}$$

where $\mathbf{I}_n$ denotes the n-dimensional identity matrix.

It is worth remarking that the variance σ_{sync}^2 of the synchronization error is typically larger than the variance of the angle measurement error σ_θ^2. Therefore, the correlation matrix R_e is not diagonal (i.e., the different cumulative error terms are not independent). Proper modeling of this fact (as we did here) is the crucial point that allows to tackle the problem of poorly synchronized PMUs in an effective way.

III. Weighted least square state estimation

We adopt the classical approach in which the power state estimation problem is posed as a *weighted least square problem* [16], [27]. We therefore consider the optimization problem

$$\left(\hat{U}, \hat{\theta}_u\right) = \arg\min_{U,\theta_u} \; J(U, \theta_u) \tag{4}$$

where the cost function $J(U, \theta_u)$ is defined according to the measurement model of Section II-B, as

$$J(U, \theta_u) = z^T R_e^{-1} z, \tag{5}$$

with z defined as

$$z = \begin{bmatrix} U^{(m)} - U \\ \theta_u^{(m)} - \theta_u \\ I^{(m)} - I(U, \theta_u) \\ \theta_i^{(m)} - \theta_i(U, \theta_u) \end{bmatrix}. \tag{6}$$

The functions $I(U, \theta_u)$ and $\theta_i(U, \theta_u)$ in (6) derive from the grid model presented in Section II-A and are nonlinear functions of the decision variables U and θ_u. Therefore the optimization problem (4) is in general non-convex, and can be difficult to tackle via standard iterative algorithms.

As proposed in other state estimation approaches based on PMUs, we introduce a rectangular representation of the state, so that we will be able to introduce a convenient linearized measurement model and, ultimately, a convenient closed form solution to (4). Let us define the complex valued measurements

$$\begin{aligned} u_v^{(m)} &= U_v^{(m)} \exp\left[j\theta_{u_v}^{(m)}\right] \\ i_v^{(m)} &= I_v^{(m)} \exp\left[j\theta_{i_v}^{(m)}\right]. \end{aligned} \tag{7}$$

The measurement model can be rewritten as a linear measurement model in the form

$$m = H\,x \,+\, \eta \tag{8}$$

in which[1]

$$m = \begin{bmatrix} \Re(u^{(m)}) \\ \Im(u^{(m)}) \\ \Re(i^{(m)}) \\ \Im(i^{(m)}) \end{bmatrix}, \; x = \begin{bmatrix} \Re(u) \\ \Im(u) \end{bmatrix}, \; \eta = \begin{bmatrix} \Re(e_u) \\ \Im(e_u) \\ \Re(e_i) \\ \Im(e_i) \end{bmatrix},$$

where the vectors e_u and e_i are defined element-wise as

$$e_{u_v} = u_v - u_v^{(m)}, \qquad e_{i_v} = i_v - i_v^{(m)},$$

and

$$H = \begin{bmatrix} \mathbf{I}_n & 0 \\ 0 & \mathbf{I}_n \\ \Re(L) & -\Im(L) \\ \Im(L) & \Re(L) \end{bmatrix}. \tag{9}$$

Problem (4) can now be equivalently reformulated as

$$\min_x \;\; (m - Hx)^T [R_\eta(x)]^{-1} (m - Hx) \tag{10}$$

where $R_\eta(x) = E[\eta\eta^T]$.

The nonlinearity of the original model is now entirely contained in the error covariance matrix $R_\eta(x)$, which depends on x and makes the problem (10) hard to solve in general. This matrix is not diagonal, because both magnitude and phase errors will be reprojected into both real and imaginary components in the error vector η of the linear model (8).

A suitable approximation for R_η (that maintains the correct modeling of the time synchronization errors between PMUs) has been obtained in the Appendix, and takes the form

$$R_\eta \approx \begin{bmatrix} \Sigma_{\Re(u)} & \Sigma_{\Re(u)\Im(u)} & \Sigma_{\Re(u)\Re(i)} & \Sigma_{\Re(u)\Im(i)} \\ \Sigma_{\Im(u)\Re(u)} & \Sigma_{\Im(u)} & \Sigma_{\Im(u)\Re(i)} & \Sigma_{\Im(u)\Im(i)} \\ \Sigma_{\Re(i)\Re(u)} & \Sigma_{\Re(i)\Im(u)} & \Sigma_{\Re(i)} & \Sigma_{\Re(i)\Im(i)} \\ \Sigma_{\Im(i)\Re(u)} & \Sigma_{\Im(i)\Im(u)} & \Sigma_{\Im(i)\Re(i)} & \Sigma_{\Im(i)} \end{bmatrix} \tag{11}$$

where the matrices $\Sigma_{\Re(u)}$, $\Sigma_{\Re(u)\Im(u)}$, $\Sigma_{\Re(u)\Re(i)}$, $\Sigma_{\Re(u)\Im(i)}$, $\Sigma_{\Im(u)}$, $\sigma_{\Im(u)\Re(i)}$, $\Sigma_{\Im(u)\Im(i)}$, $\Sigma_{\Re(i)}$, $\Sigma_{\Re(i)\Im(i)}$, $\Sigma_{\Im(i)}$, $\in \mathbb{R}^{n\times n}$ are all diagonal matrices whose v-th diagonal elements depend on the measurements and are given by

$$\begin{aligned} \left[\Sigma_{\Re(u)}\right]_{vv} &= \sigma_U^2 \cos^2(\theta_{u_v}^{(m)}) + \sigma_\theta^2 (U_v^{(m)})^2 \sin^2(\theta_{u_v}^{(m)}) \\ &\quad + \sigma_{\text{sync}}^2 (U_v^{(m)})^2 \sin^2(\theta_{u_v}^{(m)}) \\ \left[\Sigma_{\Im(u)}\right]_{vv} &= \sigma_U^2 \sin^2(\theta_{u_v}^{(m)}) + \sigma_\theta^2 (U_v^{(m)})^2 \cos^2(\theta_{u_v}^{(m)}) \\ &\quad + \sigma_{\text{sync}}^2 (U_v^{(m)})^2 \cos^2(\theta_{u_v}^{(m)}) \\ \left[\Sigma_{\Re(i)}\right]_{vv} &= \sigma_I^2 \cos^2(\theta_{i_v}^{(m)}) + \sigma_\theta^2 (I_v^{(m)})^2 \sin^2(\theta_{i_v}^{(m)}) \\ &\quad + \sigma_{\text{sync}}^2 (I_v^{(m)})^2 \sin^2(\theta_{i_v}^{(m)}) \\ \left[\Sigma_{\Im(i)}\right]_{vv} &= \sigma_I^2 \sin^2(\theta_{i_v}^{(m)}) + \sigma_\theta^2 (I_v^{(m)})^2 \cos^2(\theta_{i_v}^{(m)}) \\ &\quad + \sigma_{\text{sync}}^2 (I_v^{(m)})^2 \cos^2(\theta_{i_v}^{(m)}) \\ \left[\Sigma_{\Re(u)\Im(u)}\right]_{vv} &= \left(\sigma_U^2 + (\sigma_\theta^2 + \sigma_{\text{sync}}^2)(U_v^{(m)})^2\right) \cdot \\ &\quad \cdot \sin(\theta_{u_v}^{(m)}) \cos(\theta_{u_v}^{(m)}) \\ \left[\Sigma_{\Re(i)\Im(i)}\right]_{vv} &= \left(\sigma_I^2 + (\sigma_\theta^2 + \sigma_{\text{sync}}^2)(I_v^{(m)})^2\right) \cdot \\ &\quad \cdot \sin(\theta_{i_v}^{(m)}) \cos(\theta_{i_v}^{(m)}) \end{aligned}$$

[1] The notations $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary part of a vector, respectively.

$$\left[\Sigma_{\Re(u)\Re(i)}\right]_{vv} = \sigma^2_{\text{sync}} U_v^{(m)} I_v^{(m)} \sin(\theta_{u_v}^{(m)}) \sin(\theta_{i_v}^{(m)})$$
$$\left[\Sigma_{\Re(u)\Im(i)}\right]_{vv} = -\sigma^2_{\text{sync}} U_v^{(m)} I_v^{(m)} \sin(\theta_{u_v}^{(m)}) \cos(\theta_{i_v}^{(m)})$$
$$\left[\Sigma_{\Im(u)\Re(i)}\right]_{vv} = -\sigma^2_{\text{sync}} U_v^{(m)} I_v^{(m)} \cos(\theta_{u_v}^{(m)}) \sin(\theta_{i_v}^{(m)})$$
$$\left[\Sigma_{\Im(u)\Im(i)}\right]_{vv} = \sigma^2_{\text{sync}} U_v^{(m)} I_v^{(m)} \cos(\theta_{u_v}^{(m)}) \cos(\theta_{i_v}^{(m)}).$$

Notice that the value of these elements in the proposed approximation depends on the value of the measurements, and does not depend on the state x. Therefore, the optimal solution x^* of problem (10) can be written in closed form as

$$x^* = \left(H^T R_\eta^{-1} H\right)^{-1} H^T R_\eta^{-1} m. \tag{12}$$

IV. Problem decomposition

To compute the optimal solution x^* directly as in (12), one needs to know the entire measurement vector and the entire matrices H and R_η. Moreover, the matrix $H^T R_\eta^{-1} H$ need to be inverted. For a large-scale power distribution grid, such computation imposes a limit to the dimension of the matrix H, and hence on the number of measurements that can be efficiently processed to obtain a real-time state estimate, given the available computation time between subsequent measurement instants. To reduce the computational complexity of the problem we adopt the multi-area decomposition approach proposed in [23], and also advocated in the IEEE standard C37.118-2005 [29]. Following this approach, we assume that the grid is partitioned into non-overlapping sub-areas, each of them provided with a data processor with computational and communication capabilities. Each processor

- can collect raw data measurement from the PMUs belonging to its sub-area;
- has a local knowledge of the grid topology, namely, it knows the grid topology pertaining its sub-area and it also knows which power lines connect the sub-area with other adjacent sub-areas (called neighboring areas);
- can communicate with the monitors of the neighboring sub-areas.

. We aim to design a distribute state estimation algorithm so that, based on the available information, each processor ultimately obtains the optimal estimate of the portion of state related to its sub-area.

In order to derive a formal notation for the problem decomposition, let the set of buses $\mathcal{V}$ be partitioned into the s sub-areas $\mathcal{A}_1, \dots, \mathcal{A}_s$, where $|\mathcal{A}_j| = n_j$, with $\sum_{j=1}^s n_j = n$. Without loss of generality, assume that the bus indices are conveniently ordered, so that the vectors i and u are partitioned as

$$i = \begin{bmatrix} i_{\mathcal{A}_1} \\ \vdots \\ i_{\mathcal{A}_s} \end{bmatrix}, \qquad u = \begin{bmatrix} u_{\mathcal{A}_1} \\ \vdots \\ u_{\mathcal{A}_s} \end{bmatrix}.$$

Two sub-areas $\mathcal{A}_h$ and $\mathcal{A}_k$ are said to be neighboring sub-areas if there is a power line connecting them, i.e. if there exist $v \in \mathcal{A}_h$ and $v' \in \mathcal{A}_k$ such that the edge (v, v') or the edge (v', v) belong to $\mathcal{E}$. In such case, we say that sub-area $\mathcal{A}_h$ is a neighbor of sub-area $\mathcal{A}_k$ and vice-versa. For $h \in \{1, \dots, s\}$ we denote by $\mathcal{N}_h \subseteq \{1, \dots, s\}$ the set of the indices of the neighbors of $\mathcal{A}_h$.

The above partition also induces a corresponding block structure for the Laplacian L, i.e., $L = [L_{hk}]$, $h, k = 1, \dots, s$, such that

$$i_{\mathcal{A}_h} = \sum_{k=1}^{s} L_{hk} u_{\mathcal{A}_k} = \sum_{k \in \mathcal{N}_h \cup \{h\}} L_{hk} u_{\mathcal{A}_k}.$$

The area $\mathcal{A}_h$ is assigned to monitor h which

1) knows the matrices L_{hh}, $\{L_{hk}\}_{k \in \mathcal{N}_h}$;
2) collects the measurements $U_{\mathcal{A}_h}^{(m)}$, $\theta_{u_{\mathcal{A}_h}}^{(m)}$, $I_{\mathcal{A}_h}^{(m)}$ and $\theta_{i_{\mathcal{A}_h}}^{(m)}$;
3) knows the variances $\sigma_U^2, \sigma_I^2, \sigma_\theta^2$, and σ^2_{sync}.

The linear model (8) can therefore be expressed in the corresponding partitioned form. Specifically

$$\begin{aligned}
&\Re(u_{\mathcal{A}_h}^{(m)}) = \Re(u_{\mathcal{A}_h}) + \Re(e_{u_{\mathcal{A}_h}}) \qquad (13)\\
&\Im(u_{\mathcal{A}_h}^{(m)}) = \Im(u_{\mathcal{A}_h}) + \Im(e_{u_{\mathcal{A}_h}})\\
&\Re(i_{\mathcal{A}_h}^{(m)}) =\\
&\sum_{k \in \mathcal{N}_h \cup \{h\}} \left(\Re(L_{hk})\Re(u_{\mathcal{A}_k}) - \Im(L_{hk})\Im(u_{\mathcal{A}_k})\right) + \Re(e_{i_{\mathcal{A}_h}})\\
&\Im(i_{\mathcal{A}_h}^{(m)}) =\\
&\sum_{k \in \mathcal{N}_h \cup \{h\}} \left(\Im(L_{hk})\Re(u_{\mathcal{A}_k}) + \Re(L_{hk})\Im(u_{\mathcal{A}_k})\right) + \Im(e_{i_{\mathcal{A}_h}})
\end{aligned}$$

Now, let

$$m_{\mathcal{A}_h} = \begin{bmatrix} \Re(u_{\mathcal{A}_h}^{(m)}) \\ \Im(u_{\mathcal{A}_h}^{(m)}) \\ \Re(i_{\mathcal{A}_h}^{(m)}) \\ \Im(i_{\mathcal{A}_h}^{(m)}) \end{bmatrix}, \; x_{\mathcal{A}_h} = \begin{bmatrix} \Re(u_{\mathcal{A}_h}) \\ \Im(u_{\mathcal{A}_h}) \end{bmatrix}, \; \eta_{\mathcal{A}_h} = \begin{bmatrix} \Re(e_{u_{\mathcal{A}_h}}) \\ \Im(e_{u_{\mathcal{A}_h}}) \\ \Re(e_{i_{\mathcal{A}_h}}) \\ \Im(e_{i_{\mathcal{A}_h}}) \end{bmatrix}.$$

Then we can write that

$$m_{\mathcal{A}_h} = \tilde{H}_{hh} x_{\mathcal{A}_h} + \sum_{k \in \mathcal{N}_h} \tilde{H}_{hk} x_{\mathcal{A}_k} + \eta_{\mathcal{A}_h}$$

where the matrices $\tilde{H}_{hh} \in \mathbb{R}^{4n_h \times 2n_h}$, $\tilde{H}_{hk} \in \mathbb{R}^{4n_h \times 2n_k}$ are obtained by reordering the elements of the matrix H defined in (9), according to the Equations in (13).

Finally, observe that the matrix R_η can be rearranged into a block diagonal matrix $\operatorname{diag}\left(R_{\eta_{\mathcal{A}_1}}, \dots, R_{\eta_{\mathcal{A}_n}}\right)$ where the matrix $R_{\eta_{\mathcal{A}_h}} \in \mathbb{R}^{4n_h \times n_h}$ is the covariance matrix associated to the noise $\eta_{\mathcal{A}_h}$. By introducing the function

$$\beta_h\left(x_{\mathcal{A}_h}, \{x_{\mathcal{A}_k}\}_{k \in \mathcal{N}_h}\right) = m_{\mathcal{A}_h} - \tilde{H}_{hh} x_{\mathcal{A}_h} - \sum_{k \in \mathcal{N}_h} \tilde{H}_{hk} x_{\mathcal{A}_k},$$

we have that the problem (10) is equivalent to the problem

$$\min_{x_{\mathcal{A}_1}, \dots, x_{\mathcal{A}_s}} \sum_{h=1}^{s} J_h\left(x_{\mathcal{A}_h}, \{x_{\mathcal{A}_k}\}_{k \in \mathcal{N}_h}\right) \tag{14}$$

where

$$J_h\left(x_{\mathcal{A}_h}, \{x_{\mathcal{A}_k}\}_{k \in \mathcal{N}_h}\right) = \beta_h^T R_{\eta_{\mathcal{A}_h}} \beta_h.$$

In next section we propose two distributed algorithms to solve optimization problems in the separable quadratic form (14), and that can therefore be used in our application.

Remark. *The case of a peer-to-peer architecture is one special case of the architecture that we just described. Such special case can be easily obtained just by considering* *"atomic" areas, each one encompassing one single PMU. We therefore have $n_j = 1$ for all areas, and the set of neighbors of each area simply corresponds to the neighbor buses in the power distribution grid.*

V. Distributed optimization algorithms

The minimization problem formulated in (14) belongs to a common setup in distributed optimization that we review here in a general form, in order to reduce the complexity of the notation to a minimum.

Consider a network with set of nodes $V = \{1, \ldots, s\}$ and fixed undirected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{N}_i$ denote the set of neighbors of node i, that is, $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$. The graph $\mathcal{G}$ is assumed to be connected. Consider the minimization of a cost function in the form

$$\min_x \sum_{i=1}^{s} J_i(x) \tag{15}$$

where each J_i is a convex function of x and it is known only to node i. Assume problem in (15) admits a unique solution x^*. In this section we consider problems as in (15) with a specific structure, that is a *partition-based structure*. Let the vector x (and, similarly, the vectore x^*) be partitioned as

$$x = \left[x_1^T, \ldots, x_s^T\right]^T$$

where, for $i \in \{1, \ldots, s\}$, $x_i \in \mathbb{R}^{m_i}$ for some $m_i \in \mathbb{N}$ such that $\sum_{i=1}^{s} m_i = N$. The sub-vector x_i represents the relevant information at node i, referred to, hereafter, as the state of node i. Additionally, let us assume that the each objective functions J_i depends only on the state of node i and on its neighbors, that is, on x_i and $\{x_j\}_{j\in\mathcal{N}_i}$, where x_i and x_j are the portions of the state vector x that correspond to node i and to each neighbor node j, respectively. Then the problem we aim at solving distributively is in the form

$$\min_x \sum_{i=1}^{s} J_i(x_i, \{x_j\}_{j\in\mathcal{N}_i}). \tag{16}$$

Because of the specific least square estimation problem that we are considering, the functions J_i has a specific quadratic form, and therefore we further restrict to the case where

$$\begin{aligned} &J_i(x_i, \{x_j\}_{j\in\mathcal{N}_i}) = \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (17) \\ &\left(q_i - A_{ii}x_i - \sum_{j\in\mathcal{N}_i} A_{ij}x_j\right)^T Q_i \left(q_i - A_{ii}x_i - \sum_{j\in\mathcal{N}_i} A_{ij}x_j\right). \end{aligned}$$

To solve (16), and therefore to ultimately solve our state estimation problem, we now propose two iterative algorithms.

A. ADMM-based algorithm

The method we propose in this subsection is a partition-based version of the classical ADMM method which exploits the equivalence between problem in (16) and the following problem

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{s} J_i(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}) \\ \text{subject to} \quad & x_i^{(i)} = z_i^{(i,j)};\ x_j^{(i)} = z_j^{(i,j)} \\ & x_i^{(i)} = z_i^{(j,i)};\ x_j^{(i)} = z_j^{(j,i)}, \qquad \forall\, j \in \mathcal{N}_i, \end{aligned} \tag{18}$$

where with the notation $x_j^{(i)}$ we denote a copy of state x_j stored in memory by node i, and where the z's are auxiliary variables that are introduced by the ADMM algorithm. Observe that the connectedness of the graph $\mathcal{G}$ and the presence of the bridge variables $z's$ ensures that the optimal solution of (18) is given by $x_i^{(i)} = x_i^*$ and $x_j^{(i)} = x_j^*$.

The redundant constraints added in problem (18) with the respect to problem (16), allow to find the optimal solution through a distributed, iterative, partition-based implementation which optimizes the standard augmented Lagrangian defined, for $\rho > 0$, as

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{s} \Bigg\{ & J_i(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}) + \sum_{j\in\mathcal{N}_i} \Big[\lambda_i^{(i,j)}\left(x_i^{(i)} - z_i^{(i,j)}\right) \\ & +\lambda_j^{(i,j)}\left(x_j^{(i)} - z_j^{(i,j)}\right)\Big] + \sum_{j\in\mathcal{N}_i}\Big[\mu_i^{(i,j)}\left(x_i^{(i)} - z_i^{(j,i)}\right) \\ & +\mu_j^{(i,j)}\left(x_j^{(i)} - z_j^{(j,i)}\right)\Big] + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|x_i^{(i)} - z_i^{(i,j)}\|^2 \\ & +\|x_j^{(i)} - z_j^{(i,j)}\|^2 + \|x_i^{(i)} - z_i^{(j,i)}\|^2 + \|x_j^{(i)} - z_j^{(j,i)}\|^2\Big]\Bigg\}. \end{aligned}$$

At each iteration of the algorithm, node i, $i \in \{1, \ldots, s\}$, alternates dual ascent step on the Lagrange multipliers $\lambda_i^{(i,j)}$, $\{\lambda_j^{(i,j)}\}_{j\in\mathcal{N}_i}$ and $\mu_i^{(i,j)}$, $\{\mu_j^{(i,j)}\}_{j\in\mathcal{N}_i}$, with minimization steps on the variables $x_i^{(i)}$, $\{x_j^{(i)}\}_{j\in\mathcal{N}_i}$ and $z_i^{(i,j)}$, $\{z_j^{(i,j)}\}_{j\in\mathcal{N}_i}$.

However, for the case where the functions $J_i's$ have the particular quadratic structure illustrated in (17), these optimization steps can be greatly simplified. Indeed in this case the partition-based ADMM algorithm reduces to a linear algorithm requiring, during each iteration of its implementation, only one communication round involving the $x_i^{(i)}$, $\{x_j^{(i)}\}_{j\in\mathcal{N}_i}$, $i \in \{1, \ldots, s\}$, variables. To show that, it is convenient to introduce the following compact notation. Consider node i and, without loss of generality, assume $\mathcal{N}_i = \left\{j_1, \ldots, j_{|\mathcal{N}_i|}\right\}$. Then let

$$X^{(i)} = \begin{bmatrix} x_i^{(i)} \\ \left\{x_j^{(i)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix},$$

$$A_i = \left[A_{ii}\ A_{ij_1}\ \ldots\ A_{ij_{|\mathcal{N}_i|}}\right],$$

$$M_i = \text{diag}\left\{|\mathcal{N}_i|\, I_{m_i}, I_{m_{j_1}}, \ldots, I_{m_{j_{|\mathcal{N}_i|}}}\right\}$$

Additionally we introduce the following auxiliary variables,

$$G^{(i)} = \begin{bmatrix} G_i^{(i)} \\ G_{j_1}^{(i)} \\ \vdots \\ G_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix},\ F^{(i)} = \begin{bmatrix} F_i^{(i)} \\ F_{j_1}^{(i)} \\ \vdots \\ F_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix},\ B^{(i)} = \begin{bmatrix} B_i^{(i)} \\ B_{j_1}^{(i)} \\ \vdots \\ B_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix}$$

where $G_i^{(i)}, F_i^{(i)}, B_i^{(i)} \in \mathbb{R}^{m_i}$ and $G_{j_h}^{(i)}, F_{j_h}^{(i)}, B_{j_h}^{(i)} \in \mathbb{R}^{m_{j_h}}$. It turns out that $A_i \in \mathbb{R}^{r_i \times \gamma_i}$, $M_i \in \mathbb{R}^{\gamma_i \times \gamma_i}$ and $G^{(i)}, F^{(i)}, B^{(i)} \in \mathbb{R}^{\gamma_i}$, where $\gamma_i = m_i + \sum_{h=1}^{|\mathcal{N}_i|} m_{j_h}$.

The *partition-based ADMM algorithm* for quadratic functions is formally described as follows. The standing assumption is that all the matrices $A_i^T Q_i\, A_i + M_i$, $i \in \{1, \ldots, n\}$ are invertible.

Processor states: For $i \in \{1, \dots, s\}$, node i stores a copy of the variables $X^{(i)}, G^{(i)}, F^{(i)}, B^{(i)}$.

Initialization: Every node initializes the variables it stores in memory to 0.

Transmission iteration: For $t \in \mathbb{N}$, at the start of the t-th iteration of the algorithm, node i transmits to node j, $j \in \mathcal{N}_i$, its estimates $x_i^{(i)}(t)$, $x_j^{(i)}(t)$. It also gathers the t-th estimates of its neighbors, $x_j^{(j)}(t)$, $x_i^{(j)}(t)$, $j \in \mathcal{N}_i$.

Update iteration: For $t \in \mathbb{N}$, node i, $i \in \{1, \dots, s\}$, based on the information received from its neighbors, perform the following actions in order:

1) it computes $G^{(i)}(t+1)$ by setting

$$G_i^{(i)}(t) = \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \left(x_i^{(i)}(t) - x_i^{(j)}(t) \right)$$

$$G_{j_h}^{(i)}(t) = \frac{\rho}{2} \left(x_{j_h}^{(i)} - x_{j_h}^{(j_h)} \right), \quad 1 \le h \le |\mathcal{N}_i|$$

2) it computes $F^{(i)}(t+1)$ by

$$F^{(i)}(t+1) = F^{(i)}(t) + G^{(i)}(t)$$

3) it computes $B^{(i)}(t+1)$ by

$$B^{(i)}(t+1) = 2\rho M_i X^{(i)}(t) - G^{(i)}(t+1) - 2F^{(i)}(t+1)$$

4) it updates $X^{(i)}$ as follows

$$X^{(i)}(t+1) =$$
$$\left[A_i^T Q_i \, A_i + M_i \right]^{-1} \left[A_i^T Q_i z_i + \frac{1}{2} B^{(i)}(t+1) \right]$$

The following proposition characterizes the performance of the above algorithm.

Proposition V.1. *Consider the partition-based ADMM algorithm described above. Let ρ be any real number. Assume that the matrices $A_i^T Q_i \, A_i + M_i$, $i \in \{1, \dots, s\}$, are invertible. Then the trajectory $t \to \left\{ X^{(i)}(t) \right\}$ converge exponentially to the optimal solution, namely, for $i \in \{1, \dots, n\}$, $x_j^{(i)}(t) \to x_j^*$ for all $j \in \mathcal{N}_i$ and, in particular,*

$$x_i^{(i)}(t) \to x_i^*.$$

Proof: The proof can be found in [28]. ■

Remark. *We point out that it is be possible to provide a simplified a version of the partition-based ADMM algorithm, where two monitors communicate with each other only the information related to those nodes which are on the boundary of their sub-areas of interest. To be more precise, monitor i receives from monitor j only those components of $x_j^{(j)}$ (resp. of $x_i^{(j)}$) such that the corresponding terms in A_{ij} (resp. in A_{ji}) are different from zero. This leads to a vector $X^{(i)}$ which is smaller in size, since monitor i keeps in memory only a copy of the states of the nodes of the other sub-regions which are on the boundary with sub-region A_i. In turn, also the vectors $G^{(i)}, F^{(i)}, B^{(i)}$ and the matrices M_i, A_i result to be smaller in size.*

It is worth mentioning that this approach has been taken in [10]. In this paper, for the sake of the notational simplicity, and because we are mostly interested in cases in which the size of each area is small (possibly just one agent), we have preferred not to provide the details of this implementation aspect.

B. Jacobi-like algorithm

The method we propose in this subsection is inspired by the Jabobi technique used to iteratively solve systems of linear equations [31]. In the sequel we refer to this method as the *Jacobi-like algorithm*. The algorithm is formally described as follows. The standing assumption is that the matrices $A_{ii}^T Q_i A_{ii}$, $i \in \{1, \dots, s\}$ are all invertible.

Processor states: For $i \in \{1, \dots, s\}$, node i stores an estimate $x_i(0) \in \mathbb{R}^{m_i}$ of its own state.

Initialization: Every node initializes its estimate to an arbitrary value.

Transmission iteration: For $t \in \mathbb{N}$, at the start of the t-th iteration of the algorithm, node i transmits its estimate $x_i(t)$ to all its neighbors. It also gathers the t-th estimates of its neighbors, $x_j(t)$, $j \in \mathcal{N}_i$.

Update iteration: For $t \in \mathbb{N}$, node i, $i \in \{1, \dots, s\}$, based on the information received from its neighbors, updates its estimate as follows

$$x_i(t+1) = \underset{x_i}{\operatorname{argmin}} \;\; J_i \left(x_i; \; \{x_j(t)\}_{j \in \mathcal{N}_i} \right)$$
$$= \left(A_{ii}^T Q_i A_{ii} \right)^{-1} A_{ii}^T Q_i \left(z_i - \sum_{j \in \mathcal{N}_i} A_{ij} x_j(t) \right)$$

To establish the convergence properties of the Jacobi-like algorithm it is convenient to introduce the following block matrix $K = [K_{ij}]$, $i, j = 1, \dots, s$, where $K_{ij} \in \mathbb{R}^{m_i \times m_j}$ is defined as

$$K_{ij} = \begin{cases} A_{ii}^T Q_i A_{ii} & \text{if } j = i \\ A_{ii}^T Q_i A_{ij} & \text{if } j \in \mathcal{N}_i, \; j \neq i \\ 0 & \text{if } j \notin \mathcal{N}_i \end{cases}$$

In the following Proposition, by $\operatorname{diag}\{K\}$ we denote the block diagonal matrix having in the diagonal the blocks $K_{11}, \dots, K_{ss}$. We have the following result.

Proposition V.2. *Assume the spectral radius of the matrix $(diag\,\{K\})^{-1}(K - diag\,\{K\})$ is strictly less than one, i.e., all its eigenvalues are strictly inside the unitary circle. Then, there exists $\bar{x} \in \mathbb{R}^N$, such that the trajectory $t \to x(t)$ generated by the Jacobi-like algorithm converges exponentially to $\bar{x}$. If the matrix K is invertible then*

$$\bar{x} = K^{-1} \bar{z},$$

where $\bar{z} = \left[(A_{11}^T Q_1 z_1)^T, \dots, (A_{ss}^T Q_s z_s)^T \right]^T$.

Proof: Observe that the updating step can be written as

$$x(t+1) = (\operatorname{diag}\{K\})^{-1} \left(\bar{z} - (K - \operatorname{diag}\{K\}) x(t) \right) \quad (19)$$

where (19) represents the standard iteration of the Jacobi method. The result established in the Proposition follows from the classical results on the Jacobi method [31]. ■

Some remark are now in order.

Remark. *In general the state $\bar{x}$ is different from the optimal state x^*, i.e., the Jacobi-like algorithm does not converge to the optimal solution. However we will see in Section VI that, when, applied to Problem* (14)*, the Jacobi-like algorithm converges to an estimate $\bar{x}$ which is very close to x^*.*

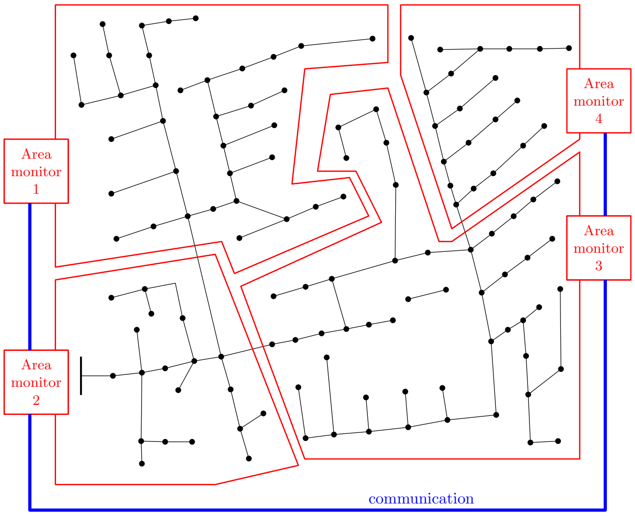

Fig. 2. IEEE test feeder 123 divided into four non-overlapping areas.

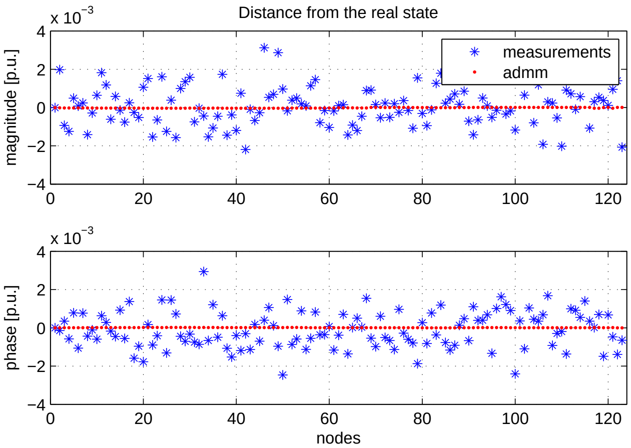

Fig. 3. Residual error at the steady state of the ADMM algorithm, compared to the raw measurements.

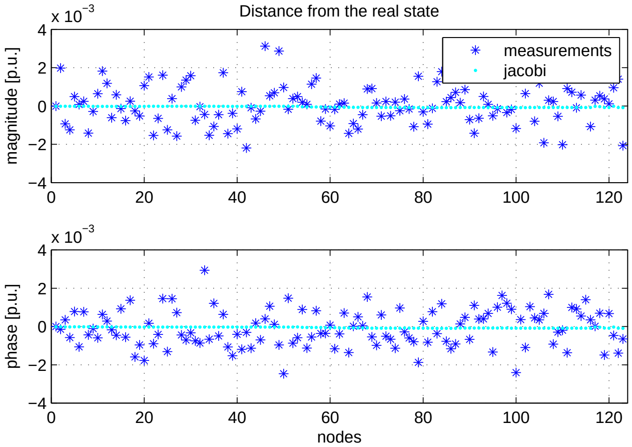

Fig. 4. Residual error at the steady state of the Jacobi-like algorithm, in the case of a 4-area architecture, compared to the raw measurements.

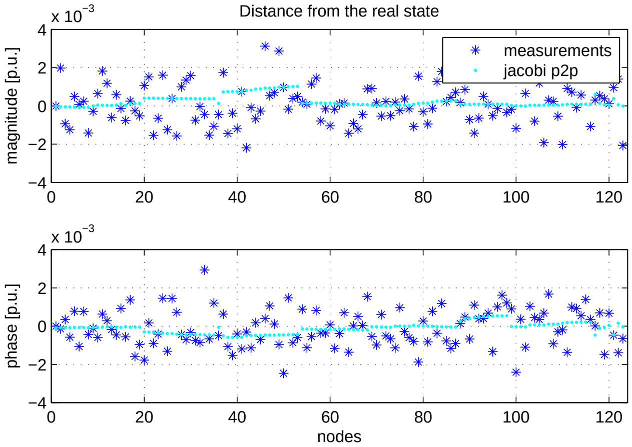

Fig. 5. Residual error at the steady state of the Jacobi-like algorithm, in the case of a peer-to-peer architecture, compared to the raw measurements.

Remark. *It is worth stressing that the computation of the update iteration of the Jacobi-like algorithm is simpler, and, in turn, less time-consuming, than the update iteration of the partition-based ADMM algorithm. This fact is emphasized and discussed, also numerically, in Section VI.*

VI. Simulation results

In this section we validate and compare the two proposed algorithms via simulations on standard medium voltage IEEE testbeds [32]. Specifically, in Subsection VI-A we provide a comparison between the performance of the *ADMM-based algorithm* presented in Section V-A and the *Jacobi-like algorithm* presented in V-B on the IEEE 123 test feeder. In Section VI-B, instead, we evaluate the effect of measurement errors on a prototypical feedback control algorithm on the IEEE 37 test feeder, and we quantify the beneficial effects obtained via the proposed state estimation approach.

A. State estimation accuracy

In order to evaluate and compare the performance of the two proposed algorithms, we adopted the IEEE 123 test feeder, whose parameters can be found in [32]. For both algorithms, we consider a 4-area architecture, in which four monitors collect and process the measurements of the PMUs of their respective area (see Figure 2), and a peer-to-peer architecture, in which all PMUs independently process their data and the data from their neighbors (i.e. areas correspond to individual buses).

We considered the following standard deviations of the measurement errors:

$$\begin{aligned}\text{voltage amplitude: } \sigma_U &= 10^{-3} U_N \text{ [Volt]}\\ \text{current amplitude: } \sigma_I &= 10^{-3} I_{\max} \text{ [A]}\\ \text{angle: } \sigma_\theta &= 10^{-3} \text{ [rad]}\\ \text{sync: } \sigma_{\text{sync}} &= 3 \cdot 10^{-3} \text{ [rad]}\end{aligned}$$

where U_N is the nominal voltage of the grid and where $I_{\max}$ is the largest current magnitude in the network, i.e., $\max_{v \in V} I_v$. These parameters are in accordance with the maximum measurement errors allowed by the IEEE standard C37.118-2005 [29], which specifies only an aggregate constraint on the measurement errors, without differentiating between magnitude and phase constraints:

$$\frac{|u_v^{(m)} - u_v|}{|u_v|} \leq 1\%, \qquad \frac{|i_v^{(m)} - i_v|}{|i_v|} \leq 1\%.$$

In the upper panel of Figures 3, 4, 5, we plotted, for each node v of the distribution test feeder, a comparison between the magnitude errors $\hat{U}_v - U_v$ and $U_v^{(m)} - U_v$, while, in the lower panel of the same figures, we plotted a comparison between the phase errors $\hat{\theta}_{u_v} - \theta_{u_v}$ and $\theta_{u_v}^{(m)} - \theta_{u_v}$, where by $\hat{U}_v$ and $\hat{\theta}_{u_v}$ we denote the steady state estimates computed by estimation algorithms. The state estimates in Figures 3 have been obtained via the ADMM-based algorithm, and do not depend on the adopted decomposition of the grid (4 macro areas or peer-to-peer), as the steady state is provably the unique optimal estimate. On the other hand, the estimates in Figures 4 and 5 have been obtained via the Jacobi-like algorithm, in the case of a 4-area architecture and of a peer-to-peer architecture, respectively. In all these cases, the measurement errors are drastically reduced by both the algorithms, which provide a consistent and accurate estimate of the node voltages. Notice that the Jacobi-like algorithm does not converge to the true optimal estimate, as a small steady state error is noticeable in the peer-to-peer case. It is however stable, as predicted by Proposition V.2, whose assumptions are all verified in this scenario.

We also consider the case in which the synchronization error is quite larger, up to

$$\text{sync: } \sigma_{\text{sync}} = 10^{-2} \text{ [rad]},$$

which is comparable to the synchronization accuracy that one could achieve even without GPS, if the PMUs are synchronized via data network synchronization protocols (see for example [33], [34] and references therein).

In Figures 6 and 7 we plotted the norm of the magnitude and phase error as a function of the standard deviation σ_{synch} of the time synchronization error between PMUs. It can be seen that the ADMM algorithm outperforms the Jacobi-like algorithm, even if they both reduce the measurement errors present in the raw data. As expected, the curves generated by the ADMM algorithm correspond to the optimal centralized estimate.

In order to analyze the computational complexity of the two proposed algorithm, we studied the evolution of the error norm $\|\hat{u}(t) - u^*\|$, where $\hat{u}(t) = \hat{U}(t)e^{j\hat{\theta}_u(t)}$ denotes the estimate computed after t iterations of the algorithms, and u^* is the optimal state estimate. We first plotted the error norm as a function of the number of iteration, in Figure 8 (for the 4-area architecture) and in Figure 9 (for the peer-to-peer architecture). As expected, the estimate $\hat{u}(t)$ generated by Jacobi like algorithm does not converge to u^*; instead, when adopting the ADMM-based algorithm, $\hat{u}(t)$ converges exponentially to u^*. Observe however that the updating step of the Jacobi-like algorithm is computationally much simpler than the corresponding iteration in the ADMM-based algorithm; for this reason, we then compared the evolution of the error norm as a function of computational time in Figure 10 (for the 4-area architecture) and in Figure 11 (for the peer-to-peer architecture). In this comparison, the Jacobi-like algorithm exhibit a faster transient, and may be preferred if the system sampling time is short, compared to the available computational capabilities at the agents.

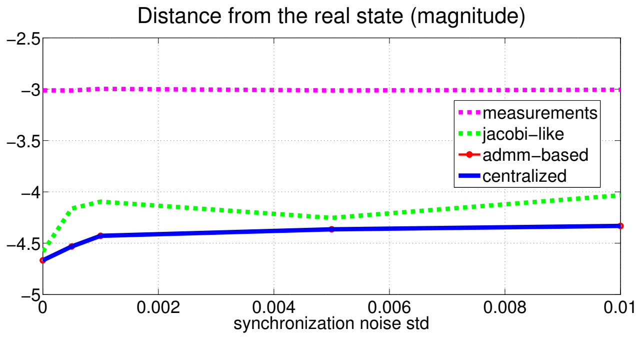

Fig. 6. Comparison of the norm of the magnitude estimation error for different values of the standard deviation σ_{sync}. Both algorithms have been reported, together with the raw measurements and the optimal centralized estimate (which corresponds to the estimate generated by the ADMM algorithm). The plot represents the average of 1000 trials, and the norm of the error has been plotted in logarithmic scale.

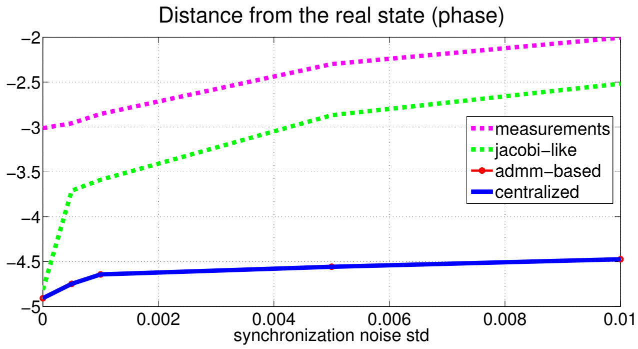

Fig. 7. Comparison of the norm of the phase estimation error for different values of the standard deviation σ_{sync}. Both algorithms have been reported, together with the raw measurements and the optimal centralized estimate (which corresponds to the estimate generated by the ADMM algorithm). The plot represents the average of 1000 trials, and the norm of the error has been plotted in logarithmic scale.

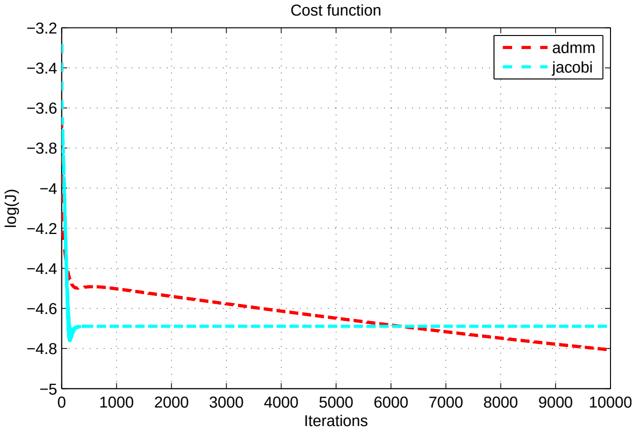

Fig. 8. Behavior of the error norm $\|\hat{u}(t) - u^*\|$ for both the ADMM and the Jacob-like estimation algorithm, in the case of a 4-area architecture, as function of the number of iterations. The norm of the error has been plotted in logarithmic scale.

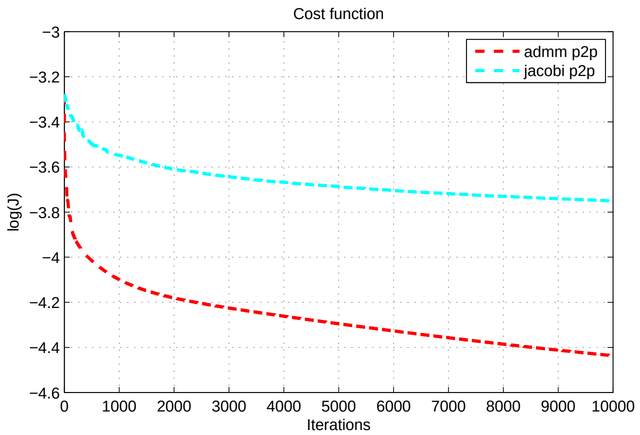

Fig. 9. Behavior of the error norm $\|\hat{u}(t) - u^*\|$ for both the ADMM and the Jacob-like estimation algorithm, in the case of a peer-to-peer architecture, as function of the number of iterations. The norm of the error has been plotted in logarithmic scale.

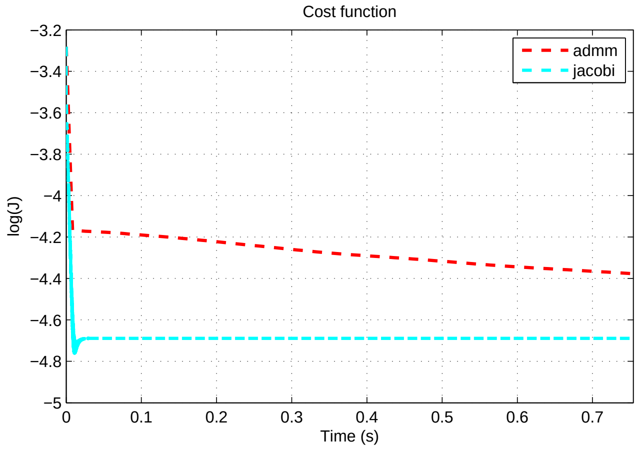

Fig. 10. Behavior of the error norm $\|\hat{u}(t) - u^*\|$ for both the ADMM and the Jacob-like estimation algorithm, in the case of a 4-area architecture, as function of time. The norm of the error has been plotted in logarithmic scale.

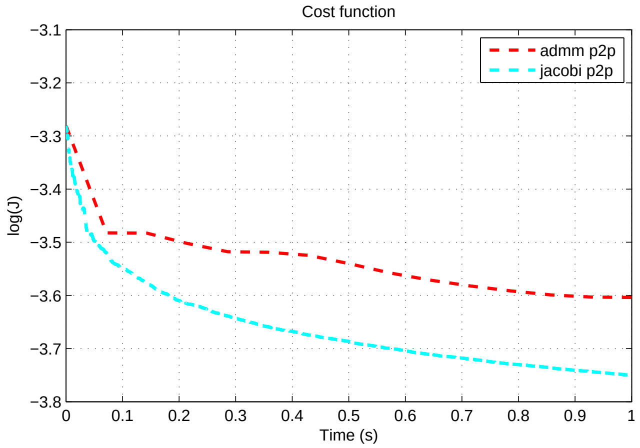

Fig. 11. Behavior of the error norm $\|\hat{u}(t) - u^*\|$ for both the ADMM and the Jacob-like estimation algorithm, in the case of a peer-to-peer architecture, as function of time. The norm of the error has been plotted in logarithmic scale.

B. Application example: Volt/VAR compensation

In order to evaluate the expected benefits of the proposed measurement layer and state estimation protocols to the smart grid operation, we investigated the effects of noisy measurements on the application presented in [15], which is a feedback control algorithm for voltage support and minimization of power distribution losses via optimal reactive power compensation.

We consider the IEEE 37 test feeder, on which 6 microgenerators have been deployed and connected to the distribution grid via power inverters (see [15] for the details of this scenario). In the mathematical analysis and in the simulations proposed in [15], noiseless voltage measurements have been considered. Here we considered the case in which voltage measurements are affected by the typical noise that characterize standard medium voltage PMUs, together with a synchronization between PMUs that has standard deviation

$$\sigma_{\text{sync}} = 10^{-2}[\text{rad}]$$

(therefore larger than the accuracy that is typically provided by GPS, and more similar to the accuracy provided by network synchronization protocols). We show in Figures 12 and 13 that, if driven by these raw data, the feedback control law is not able to reduce power distribution losses on the grid. We instead considered the case in which the raw measurements are processed according to the algorithms proposed in this paper, and the resulting state estimates are then presented to the control application. Because the control application is an iterative protocol, we need to enforce a stopping time in the execution of the state estimation algorithms. We explore two quite different scenarios:

- in the first case, represented in Figure 12, we implement the Jacobi-like algorithm, and we perform only 5 iterations of the algorithm before presenting the state estimate to the control application; because the Jacobi-like algorithm is extremely lightweight, this choice allows very short sampling time, and fast algorithm execution, also in the presence of narrow-band channels like powerline communication;
- in the second case, represented in Figure 13, we instead implement the ADMM-based algorithm, and we allow a quite larger time for computation, 50 iterations; this scenario corresponds to the case in which the control application is possibly executed at a slower pace, and a fast and reliable communication infrastructure is available.

The simulative results show how both the proposed approach are effective, enable the control algorithm to operate correctly, and ultimately allow the reduction of power distribution losses on the grid.

VII. Conclusions

We considered the motivating scenario of a smart power distribution grid, in which we assumed that a number of control applications require precise measurements of the grid state in order to optimize the grid operation, monitor it against fault and attacks, and improve grid efficiency. We modeled a practical case in which these measurements are provided by PMUs located at the grid buses, and are affected by noise. In particular, we model the measurement error that is caused by inexact synchronization of the PMUs, which is a major

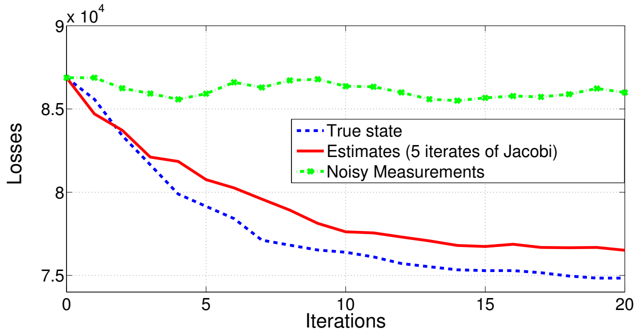

Fig. 12. Behavior of the losses minimization feedback control algorithm when driven by noiseless voltage measurments (true state), by the raw noisy measurements, and by the state estimate obtained via the Jacobi-like algorithm with a stopping time of 5 iterations at each step of the control algorithm.

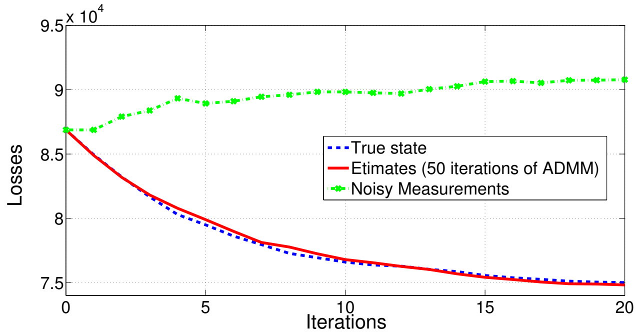

Fig. 13. Behavior of the losses minimization feedback control algorithm when driven by noiseless voltage measurments (true state), by the raw noisy measurements, and by the state estimate obtained via ADMM with a stopping time of 50 iterations at each step of the control algorithm.

issue in the deployment of low-end measurement solutions. We designed an intermediate measurement layer that collects the raw data from the PMUs, processes it, and presents an improved state estimate to the applications in the control layer. We derived two different distributed and scalable algorithms in order to process these data and to compute the optimal state estimate, and we proved their convergence. Via simulations, we showed how these algorithms are extremely effective in canceling the measurement noise, and we showed that the state estimate computed by the proposed intermediate measurement layer is accurate enough to be used by the applications in the control layer. Even if we considered only one specific control application, this result suggests that, via the proposed approach, time synchronization between PMUs in the power distribution network becomes a tractable issue. It also suggests that GPS-less PMUs could be used, if proper time synchronization algorithms are implemented via the available communication channel. This scenario is extremely motivating for further investigation in the field of distributed networked control in the smart power distribution grid.

Appendix

In this appendix we derive a suitable expression for the matrix in (11). To do so, we start by expressing the error measurements in a convenient form, starting from the knowledge of σ_U^2, σ_I^2, σ_θ^2 and σ_{sync}^2. Observe that the errors in the vector η of the linear model (8) are correlated as, in general, both magnitude and phase errors (which are uncorrelated) will be projected into both real and imaginary components. To better understand this fact consider two generic vectors

$$x = \rho\exp(j\psi) = \rho(\cos\psi + j\sin\psi),$$

and

$$\begin{aligned}\tilde{x} &= \tilde{\rho}\exp(j\tilde{\psi}) = (\rho+\delta\rho)\exp(j(\psi+\delta\psi))\\ &= (\rho+\delta\rho)(\cos(\psi+\delta\psi) + j\sin(\psi+\delta\psi))\end{aligned}$$

where $\delta\rho$ and $\delta\psi$ represent the errors affecting the exact values ρ and ψ. The quantity $\tilde{x}$ can be rewritten by exploiting classical trigonometric relations and by taking a first order approximation of the sine and cosine functions assuming $\delta\psi$ small enough, as

$$\begin{aligned}\tilde{x} \simeq\ & \rho(\cos\psi + j\sin\psi) + \delta\rho(\cos\psi + j\sin\psi)\\ & - \rho(\sin\psi - j\cos\psi)\delta\psi - \delta\rho(\sin\psi - j\cos\psi)\delta\psi.\end{aligned}$$

Discarding the second order term $\delta\rho(\sin\psi - j\cos\psi)\delta\psi$, we obtain

$$\tilde{x} \simeq x + (\delta\rho\cos\psi - \rho\sin\psi\ \delta\psi) + j(\delta\rho\sin\psi + \rho\cos\psi\ \delta\psi).$$

Let us go back to our specific context, and to the expressions in (7). According to the above approximation we have

$$\begin{aligned}\Re(e_{u_v}) &= e_{U_v}\cos\theta_{u_v}^{(m)} - (e_{\theta_{u_v}} + e_{\text{sync},v})U_v^{(m)}\sin\theta_{u_v}^{(m)}\\ \Im(e_{u_v}) &= e_{U_v}\sin\theta_{u_v}^{(m)} + (e_{\theta_{u_v}} + e_{\text{sync},v})U_v^{(m)}\cos\theta_{u_v}^{(m)}\end{aligned}$$

and

$$\begin{aligned}\Re(e_{i_v}) &= e_{I_v}\cos\theta_{i_v}^{(m)} - (e_{\theta_{i_v}} + e_{\text{sync},v})I_v^{(m)}\sin\theta_{i_v}^{(m)}\\ \Im(e_{i_v}) &= e_{I_v}\sin\theta_{i_v}^{(m)} + (e_{\theta_{i_v}} + e_{\text{sync},v})I_v^{(m)}\cos\theta_{i_v}^{(m)}.\end{aligned}$$

Consider now $\left[\Sigma_{\Re(u)}\right]_{vv}$. This term can be computed by calculating the expectation $\mathbb{E}\left[(\Re(e_{u_v}))^2\right]$, where $\theta_{u_v}^{(m)}, U_v^{(m)}$ are fixed and the noises $e_{U_v}, e_{\theta_{u_v}}, e_{\text{sync},v}$ are uncorrelated Gaussian random variables, therefore obtaining

$$\begin{aligned}\left[\Sigma_{\Re(u)}\right]_{vv} =\ & \sigma_U^2\cos^2(\theta_{u_v}^{(m)}) + \sigma_\theta^2(U_v^{(m)})^2\sin^2(\theta_{u_v}^{(m)})\\ & + \sigma_{\text{sync}}^2(U_v^{(m)})^2\sin^2(\theta_{u_v}^{(m)}).\end{aligned}$$

All the other terms of the matrix R_η are computed similarly.

References

[1] A. von Meier, D. Culler, and A. McEachern, "Micro-synchrophasors for distribution systems," in *IEEE Innovative Smart Grid Technologies (ISGT 2013)*, 2013.

[2] M. M. Albu, R. Neurohr, D. Apetrei, I. Silvas, and D. Federenciuc, "Monitoring voltage and frequency in smart distribution grids. a case study on data compression and accessibility." in *IEEE Power and Energy Society General Meeting*, 2010.

[3] L. F. Ochoa and D. H. Wilson, "Using synchrophasor measurements in smart distribution networks," in *Proc. of the 21st International Conference on Electricity Distribution*, 2011.

[4] M. Wache and D. C. Murray, "Application of synchrophasor measurements for distribution networks," in *IEEE Power and Energy Society General Meeting*, 2011.

[5] J. D. L. 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, Jun. 2010.

[6] Special issue on sensor networks and applications, *Proceedings of the IEEE*, vol. 91, no. 8, Aug. 2003.

[7] S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo, “Consensus-based distributed sensor calibration and least-square parameter identification in wsns,” *International Journal of Robust and Nonlinear Control*, vol. 20, no. 2, pp. 176–193, Jan. 2010.

[8] K. De Craemer and G. Deconinck, “Analysis of state-of-the-art smart metering communication standards,” in *Proceedings of the 5th Young Researchers Symposium*, Leuven, Belgium, Mar. 2010.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[10] V. Kekatos and G. Giannakis, “Distributed robust power system state estimation,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[11] F. Katiraei and M. R. Iravani, “Power management strategies for a microgrid with multiple distributed generation units,” *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1821–1831, Nov. 2006.

[12] M. Prodanovic, K. De Brabandere, J. Van den Keybus, T. Green, and J. Driesen, “Harmonic and reactive power compensation as ancillary services in inverter-based distributed generation,” *IET Generation, Transmission & Distribution*, vol. 1, no. 3, pp. 432–438, 2007.

[13] S. Bolognani and S. Zampieri, “A distributed control strategy for reactive power compensation in smart microgrids,” *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2818–2833, Nov. 2013.

[14] S. Bolognani, G. Cavraro, and S. Zampieri, “A distributed feedback control approach to the optimal reactive power flow problem,” in *Proceedings of Workshop on Control of Cyber-Physical Systems*, Johns Hopkins University, Baltimore (MA), USA, 2013.

[15] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri, “A distributed feedback control strategy for optimal reactive power flow with voltage constraints,” *Submitted.*, 2013. [Online]. Available: http://arxiv.org/abs/1303.7173

[16] F. C. Schweppe, J. Wildes, and D. Rom, “Power system static state estimation: Parts I, II, and III,” *IEEE Transactions Power App. Syst.*, vol. PAS-89, pp. 120–135, Jan. 1970.

[17] T. V. Cutsem and M. Ribbens-Pavella, “Critical survey of hierarchical methods for state estimation of electric power systems,” *IEEE Transactions Power App. Syst.*, vol. PAS-102, no. 10, pp. 3415–3424, Oct. 1983.

[18] S. Iwamoto, M. Kusano, and V. H. Quintana, “Hierarchical state estimation using a fast rectangular-coordinate method,” *IEEE Transactions Power Syst.*, vol. 4, no. 3, pp. 870–880, Aug. 1989.

[19] L. Zhao and A. Abur, “Multiarea state estimation using synchronized phasor measurements,” *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 611–617, May 2005.

[20] A. Gomez-Exposito, A. Abur, A. de la Villa Jean, and C. Gom’ez-Quiles, “A multilevel state estimation paradigm for smart grids,” *Proc. IEEE*, vol. 99, no. 6, pp. 952–976, Jun. 2011.

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

[22] D. M. Falcao, F. F. Wu, and L. Murphy, “Parallel and distributed state estimation,” *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 724–730, May 1995.

[23] A. J. Conejo, S. de la Torre, and M. Cañas, “An optimization approach to multiarea state estimation,” *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 213–231, Feb. 2007.

[24] R. Ebrahimian and R. Baldick, “State estimation distributed processing,” *IEEE Transactions Power Syst.*, vol. 15, no. 4, pp. 1240–1246, Nov. 2000.

[25] F. Pasqualetti, R. Carli, and F. Bullo, “Distributed estimation via iterative projections with application to power network monitoring,” *Automatica*, vol. 48, no. 5, pp. 747–758, 2012.

[26] 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, May 2007.

[27] A. Abur and A. Gómez Expósito, *Power system state estimation: theory and implementation*. New York (USA): Marcel Dekker, 2004.

[28] S. Bolognani, R. Carli, and M. Todescato, “Convergence of the partition-based ADMM for a separable quadratic cost function,” Dec. 2013, available on arXiv. [Online]. Available: http://www.dei.unipd.it/∼sbologna/papers/proof-admm.pdf

[29] K. E. Martin, D. Hamai, M. Adamiak, S. Anderson, M. Begovic, G. Benmouyal, G. Brunello, J. Burger, J. Cai, B. Dickerson, V. Gharpure, B. Kennedy, D. Karlsson, A. G. Phadke, J. Salj, V. Skendzic, J. Sperr, Y. Song, C. Huntley, B. Kasztenny, and E. Price, “Exploring the IEEE standard C37.118-2005 synchrophasors for power systems,” *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 1805–1811, Oct. 2008.

[30] E. Breitenberger, “Analogues of the normal distribution on the circle and the sphere,” *Biometrika*, vol. 50, no. 1-2, pp. 81–88, 1963.

[31] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[32] W. H. Kersting, “Radial distribution test feeders,” in *IEEE Power Engineering Society Winter Meeting*, vol. 2, Jan. 2001, pp. 908–912.

[33] R. Solis, V. Borkar, and P. R. Kumar, “A new distributed time synchronization protocol for multihop wireless networks,” in *45th IEEE Conference on Decision and Control (CDC’06)*, San Diego, December 2006, pp. 2734–2739.

[34] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri, “A randomized linear algorithm for clock synchronization in multi-agent systems,” in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC 2012)*, 2012.