A Distributed Gauss-Newton Method for Power System State Estimation

Ariana Minot, *Student Member, IEEE*, Yue M. Lu, *Senior Member, IEEE*, and Na Li, *Member, IEEE*

***Abstract*—We propose a fully distributed Gauss-Newton algorithm for state estimation of electric power systems. At each Gauss-Newton iteration, matrix-splitting techniques are utilized to carry out the matrix inversion needed for calculating the Gauss-Newton step in a distributed fashion. In order to reduce the communication burden as well as increase robustness of state estimation, the proposed distributed scheme relies only on local information and a limited amount of information from neighboring areas. The matrix-splitting scheme is designed to calculate the Gauss-Newton step with exponential convergence speed. The effectiveness of the method is demonstrated in various numerical experiments.**

***Index Terms*—Distributed algorithms, matrix splitting, state estimation.**

I. Introduction

STATE estimation plays an important role in the operation of power systems. Improving state estimation advances the capability of advanced online power dispatch, contingency analysis, frequency control, and fault diagnosis. State estimation provides a view of real-time power system conditions for the system operator to efficiently and reliably operate the power grid [1]. With the growing penetration of renewable energy, it is becoming more demanding to estimate the system state promptly and accurately. There have been increasing research efforts in developing and integrating new sensor technology to create a more advanced state estimation system. In particular, one active area of research has been the development of a Wide-Area Measurement System (WAMS) using Phasor Measurement Units (PMUs) [2].

As computational demands increase with the availability of these new measurements, there is a heightened need to develop distributed algorithms that allow each aggregate bus or control area to have its own processor for local state estimation [3]. Advantages of distributed approaches include increased robustness as well as reduction in computation, communication, and memory requirements per area since each area requires only a subset of the global information. Whether trying to achieve wide-area control between large interconnected areas of the power grid or at the micro-grid level, robust and fully-distributed state estimation will be critical [4].

There have been many research efforts in developing distributed state estimation methods for electric power systems [5]–[7]. In hierarchical distributed approaches, state estimation is carried out locally and then information is exchanged with a central processor which coordinates the local estimates to produce a global state estimate. Hierarchical approaches to state estimation in power systems are explored in [8]–[15]. One disadvantage of requiring a central coordinator is the potential for communication bottlenecks and reduced robustness.

In contrast, fully distributed state estimation utilizes neighbor-to-neighbor communication rather than relying on a central coordinator. Of recent interest have been gossip-based algorithms for fully distributed state estimation [16]–[18]. One potential shortcoming of such methods is that an estimate of the global state is required at each area. For large networks, the memory requirements can be prohibitive. Fully distributed methods requiring only local rather than global estimates per area have recently been proposed: using decomposition methods [19]–[21]; applying alternating direction method of multipliers (ADMM) [22]; and information filter-based techniques [23]. The decomposition method in [20] lacks guarantees of the convergence of the distributed state estimates to the estimates obtained by a centralized state estimator. The ADMM approach in [22] guarantees asymptotic convergence. However, possible disadvantages include the computation and storage of additional information in the form of Lagrange multipliers and complications in an asynchronous setting. The method of [23] converges in finite iterations, but the network is assumed to be acyclic and the iteration number for convergence increases linearly with the network size. For large-scale networks, asymptotically convergent methods may be preferable, especially if convergence speed scales independently of the network size.

In a broader context, there is substantial literature on developing distributed methods for general optimization problems, such as consensus-based and dual-based gradient and Newton-type methods. However, there has been a lack of work on developing distributed Newton-type methods for state estimation in power grids. Due to the nonconvexity of the power system state estimation problem, we use the Gauss-Newton method [24]. In comparison to gradient-based methods, Newton-type methods

Manuscript received April 18, 2015; revised August 08, 2015 and October 21, 2015; accepted October 23, 2015. Date of publication November 25, 2015; date of current version August 17, 2016. This work was part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. This work was also supported by the Harvard Center for Green Building and Cities. Preliminary results of this work were presented at the American Control Conference (ACC), Chicago, IL, USA, July 1–3, 2015. Paper no. TPWRS-00527-2015.

The authors are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02140 USA (e-mail:minot@fas.harvard.edu; yuelu@seas.harvard.edu; nali@seas.harvard.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2015.2497330

0885-8950 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

are advantageous with respect to convergence rate, which is usually quadratic. The difficulty is that Newton-type methods require solving a matrix inversion at each iteration. This presents a challenge for developing a distributed method since in general global information of the matrix entries is needed to invert a matrix. Furthermore, when the system size is large, solving a large matrix inversion may be time-consuming or even prohibitive. To overcome this challenge, in our work we explore the use of matrix-splitting techniques [25], [26]. This allows us to exploit inherent sparse structure in power systems in order to calculate the next Gauss-Newton iteration in a distributed way.

A similar method in [27] uses an approximate block Jacobi method, a kind of matrix splitting, for distributed state estimation. Our work is distinct in that we do not ignore the boundary terms containing information about neighboring areas. Without such boundary terms, a distributed algorithm does not require communication. Since the power network is an interconnected system, ignoring such boundary terms leaves out important information. Furthermore, theoretical guarantees for the convergence of the approximate block Jacobi iterative scheme are not provided. Our paper more fully explores some of the ideas first suggested in [27].

In our algorithm, each control area calculates the state estimates of its local buses, and communication is carried out only between neighboring areas. The contributions of this work include a new fully distributed Gauss-Newton method for power system state estimation with the following features:

1) The method incorporates both traditional SCADA (supervisory control and data acquisition) system measurements of power injections and flows, as well as PMU measurements.
2) Each control area only requires local information and a limited exchange of information with neighboring control areas in order to estimate its state, eliminating the need for a central processor.
3) Each control area only needs to hold an estimate toward its own state rather than an estimate of the global state of the system. This saves a large amount of communication and memory resources.

The principal intent of our algorithm is to be applied in a distributed computation environment with information being exchanged between different control areas across potentially large geographic distances. Our algorithm is well-suited to such a setting since it utilizes only neighbor-to-neighbor communication. However, the algorithm is also applicable for running on a parallel computing environment, where measurements are aggregated at a central location [28]. Indeed to test our algorithm, we use a computer cluster where each node in the cluster is treated as a control area. In this case, the communication time is not reflective of a geographically dispersed setting, but it gives a useful indication of the communication time requirements relative to the computation time requirements.

The paper is outlined as follows. In Section II, we present the mathematical problem statement and introduce the application of Newton's method to power system state estimation. In Section III, we present our distributed state estimation algorithm with analysis of its communication requirements. In addition, we discuss the convergence properties of our algorithm. In Section IV, numerical simulations are used to demonstrate the effectiveness of our method.

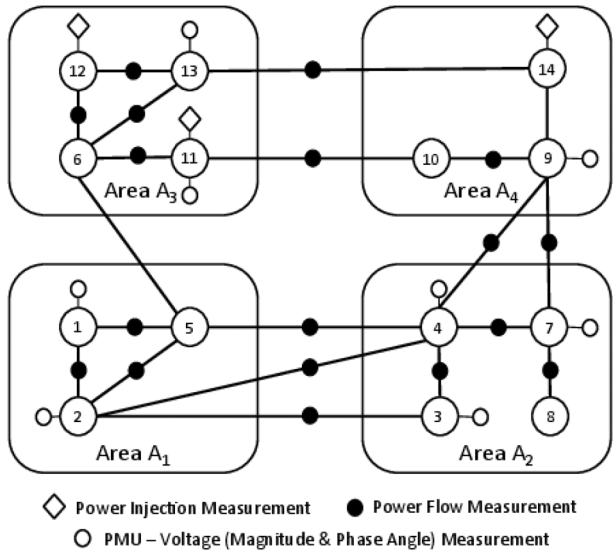


Fig. 1. Multi-area IEEE 14-bus test system partitioned into $N = 4$ control areas with location and types of measurements used [14].

Notations: We use v_k to denote the kth entry of a vector $\boldsymbol{v}$. The (i,j)th entry of a matrix $\boldsymbol{M}$ is given by M_{ij}. The transpose of a vector or matrix $\boldsymbol{X}$ is denoted by $\boldsymbol{X}^T$. The matrix inequality $\boldsymbol{M} > 0$ is to be interpreted as each element of $\boldsymbol{M}$ being positive. In contrast, we use $\boldsymbol{M} \succ 0$ to denote that $\boldsymbol{M}$ is positive definite.

II. Problem Statement and Formulation

We consider a multi-area interconnected power network, denoted by an undirected graph $(\mathcal{V}, \mathcal{E})$ with a set $\mathcal{V} \triangleq \{1, 2, \ldots, n\}$ of buses and a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of transmission lines connecting the buses. The goal of state estimation in power systems is to infer the unknown voltages (phase angle and magnitude) at each bus, $\boldsymbol{x}^T = [\theta_1\ V_1\ \ldots\ \theta_n\ V_n]$, from a set of noisy measurements of the system. The power network is partitioned into N non-overlapping regions, called control areas. Decentralized state estimation allows each control area to estimate its local state by exchanging information with neighboring control areas. Fig. 1 provides an example of the IEEE 14-bus test system under a particular control area partitioning and measurement configuration [14].

There are two typical power measurement systems. One is the traditional SCADA measurements including power flows along transmission lines and power injections at buses; the other system uses PMUs to measure the voltages and currents directly. We consider the following measurements: 1) the real and reactive power flow along the transmission line between buses i and j, denoted by $\widehat{P}_{ij}$ and $\widehat{Q}_{ij}$; 2) the real and reactive power injection at bus i, denoted by $\widehat{P}_i$ and $\widehat{Q}_i$; and 3) the voltage phase angle and magnitude at bus i, denoted by $\widehat{\theta}_i$ and $\widehat{V}_i$. Branch current phasor measurements are also available from the PMUs. As a future enhancement, the branch current measurements can be incorporated using techniques from [29]–[31] for developing hybrid state estimators that incorporate both voltage and current phasor measurements with traditional SCADA measurements. SCADA scan rates are approximately once every 2–6

seconds, whereas PMU measurements update about 30 times per second [32]. The intent of our algorithm is to use the most recent measurements available from the PMUs and from the SCADA system for each state estimation run. We consider a static setting rather than a dynamic one, treating each measurement set as a separate snapshot in time.

Due to the deployment cost of SCADA and PMU sensors, measurements of power flow, power injection, and voltage phasors are only available at a subset of the buses and transmission lines in the system. We denote the set of measurements as $\mathcal{Z}$ and the ordered vector of measurements as $\boldsymbol{z}$. The AC model relates the measurements $\boldsymbol{z}$ and the unknown state $\boldsymbol{x}$,

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ is zero-mean Gaussian random noise with covariance matrix $\boldsymbol{R}$. AC state estimation solves the following optimization problem $\arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$, where

$$f(\boldsymbol{x}) \stackrel{\text{def}}{=} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))^T \boldsymbol{W} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})). \tag{2}$$

The following assumption is made on the weighting matrix, $\boldsymbol{W}$:

Assumption 1: The weighting matrix $\boldsymbol{W}$ is diagonal with entries $W_{kk} \stackrel{\text{def}}{=} w_k > 0$.

If the inverse covariance matrix is used as the weighting matrix, this is equivalent to assuming that noise from different measurements is independent (i.e., $\boldsymbol{R}$ is diagonal). Our method can be extended to a scenario where the noise values associated with measurements from within a single control area have non-zero covariance.

Given an initial point, $\boldsymbol{x}^{(0)}$, Newton's method uses an iterative scheme to minimize $f(\boldsymbol{x})$:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \Delta\boldsymbol{x}^{(k)}. \tag{3}$$

At each iteration k, Newton's method minimizes the second-order approximation to f about $\boldsymbol{x}^{(k)}$. The Newton step, $\Delta\boldsymbol{x}^{(k)}$, is given by solving the following linear system

$$[\nabla^2 f(\boldsymbol{x}^{(k)})]\Delta\boldsymbol{x}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}), \tag{4}$$

where ∇f and $\nabla^2 f$ are the Jacobian and Hessian functions of the objective function. Since $f(\boldsymbol{x})$ is a non-convex function, we use the Gauss-Newton method which employs a positive-definite approximation to $\nabla^2 f(\boldsymbol{x})$ [24]. We denote this by $\tilde{\nabla}^2 f(\boldsymbol{x}) = \boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{W}\boldsymbol{J}(\boldsymbol{x})$, where the measurement Jacobian is given by

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial h_1(\boldsymbol{x})}{\partial \theta_1} & \frac{\partial h_1(\boldsymbol{x})}{\partial V_1} & \cdots & \frac{\partial h_1(\boldsymbol{x})}{\partial \theta_n} & \frac{\partial h_1(\boldsymbol{x})}{\partial V_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial h_m(\boldsymbol{x})}{\partial \theta_1} & \frac{\partial h_m(\boldsymbol{x})}{\partial V_1} & \cdots & \frac{\partial h_m(\boldsymbol{x})}{\partial \theta_n} & \frac{\partial h_m(\boldsymbol{x})}{\partial V_n} \end{pmatrix}. \tag{5}$$

Depending on the quality of the approximation to the Hessian, the Gauss-Newton method may not have the same quadratic convergence properties as the Newton method. In cases where $\tilde{\nabla}^2 f(\boldsymbol{x})$ approximates $\nabla^2 f(\boldsymbol{x})$ well, the Gauss-Newton method will show comparable performance to the Newton method without the computational burden of calculating the full objective function Hessian. Using the Gauss-Newton approximation, the Newton update (4) becomes

$$\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{W}\boldsymbol{J}(\boldsymbol{x})\Delta\boldsymbol{x} = -\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})), \tag{6}$$

where we suppress iteration label k to lighten notation. Solving this linear system yields the Gauss-Newton step, $\Delta\boldsymbol{x}^{(k)}$, needed to produce the next iterate in (3). For realistic power systems, this results in a large linear system, which is challenging to solve in real-time. The goal of this work is to solve the linear system in (6) in a distributed way, utilizing the sparsity of the system to ensure limited communication requirements.

III. Distributed State Estimation (DSE) Algorithm

A. Overview of DSE Algorithm

The aim of distributed state estimation is for each control area to estimate its local state. This requires a distributed solution to the linear system in (6). Rather than using standard direct matrix inversion methods, such as Gaussian elimination (i.e., LU factorization), Gauss-Jordan elimination, or Cholesky factorization [33], we develop an iterative method based on matrix splitting [25]. At every iteration, each control area exchanges information about its local state estimate with neighboring areas. The local state estimate is then updated based on information received from neighboring areas. Though direct methods exactly solve linear systems, up to rounding error, in a finite number of steps, for large systems, they often suffer from prohibitively large storage and computation requirements. In contrast, iterative methods can have a significant advantage over direct methods if they rapidly converge to a sufficiently accurate solution [33].

Control areas have access to their own local measurements and state estimates of local buses but not to system-wide measurements and estimates. Measurements at neighboring buses in other control areas will be relevant to the control area's state estimation. In particular, a measurement of power flow along a transmission line connecting two control areas requires those control areas to share their bordering bus state estimates with one another. Likewise, for a bus with a power injection connected to another control area, all of that buses neighbors' estimates must be shared with the neighboring control area.

B. Matrix Splitting for DSE Algorithm

We propose a new multi-area state estimation algorithm based on a matrix-splitting technique which allows us to calculate the Gauss-Newton step, $\Delta\boldsymbol{x}$, in a distributed way. This method is inspired by the use of matrix-splitting for developing a distributed Newton method for the Network Utility Maximization (NUM) problem of Wei *et al.* in [26]. We introduce the following notation

$$\boldsymbol{A} \stackrel{\text{def}}{=} \boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{W}\boldsymbol{J}(\boldsymbol{x}) \tag{7}$$

$$\boldsymbol{b} \stackrel{\text{def}}{=} -\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{W}(\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})). \tag{8}$$

Writing the linear system in (6) as $\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{b}$, the idea behind matrix splitting is to write $\boldsymbol{A}$ as the difference of an invertible matrix, $\boldsymbol{M}$, and a matrix $\boldsymbol{N}$, (i.e., $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$) [25]. Then for an arbitrary $\Delta\boldsymbol{x}^0 \in \mathbb{R}^n$, consider the following scheme:

$$\Delta\boldsymbol{x}^{t+1} = \boldsymbol{M}^{-1}\boldsymbol{N}\Delta\boldsymbol{x}^t + \boldsymbol{M}^{-1}\boldsymbol{b}. \tag{9}$$

The key idea is that matrix-splitting provides an iterative solution to solving (6). Since the Gauss-Newton method is itself an iterative method for minimizing the weighted least-squares

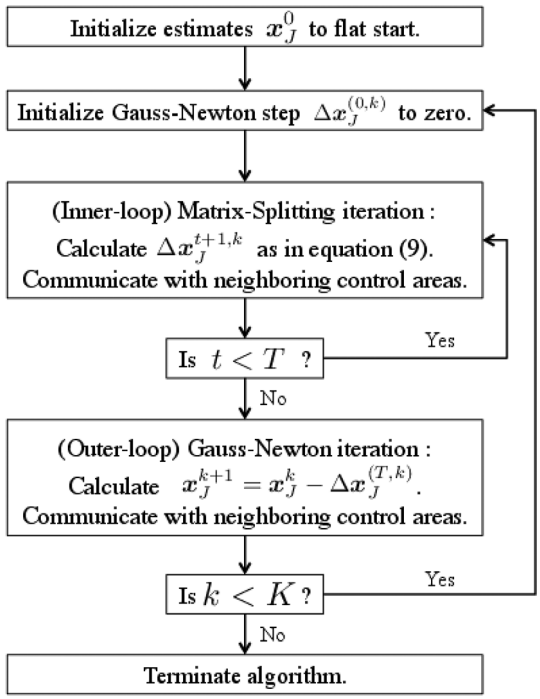


Fig. 2. Outline of proposed DSE algorithm detailing inner- and outer-loop structure for control area J.

objective function, matrix-splitting introduces an inner-loop of iterations t for each Gauss-Newton (i.e. outer-loop) iteration k. Numerical tests in Section IV demonstrate the favorable runtime of our algorithm despite this nested loop structure. A flowchart providing a practical overview of our algorithm is given in Fig. 2.

Our contribution is to design the matrix splitting so that the iterative scheme in (9) converges and is easily distributed. The sequence in (9) converges if and only if the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is strictly less than 1. For details of convergence, please see the Appendix. To facilitate distributed processing, each control area should calculate its next Gauss-Newton iterate using local information and a limited amount of communication with neighboring areas. To this end, we consider a splitting of $\boldsymbol{A}$ into a block diagonal matrix, $\boldsymbol{D}$, and an off-diagonal matrix $\boldsymbol{E}$. The entries of $\boldsymbol{D}$ correspond to local information, and the entries of $\boldsymbol{E}$ correspond to information required from other control areas.

Let the number of buses in control area A_1 be denoted n_1, and without loss of generality, let control area A_1 contain buses with node labels $S_1 = \{1, \ldots, n_1\}$. Similarly let the buses of control area A_2 have labels $S_2 = \{n_1 + 1, \ldots, n_1 + n_2\}$ and so forth for consecutive control areas. The matrix $\boldsymbol{A}$ as defined in (7) can be decomposed into the sum of a block-diagonal matrix, $\boldsymbol{D}$, and a matrix containing the remaining off-diagonal entries $\boldsymbol{E}$. The consecutive re-labeling of the node indices described above allows for all entries of $\boldsymbol{A}$ corresponding to buses within the same control area to be contained within a single diagonal block. Given a row index i of $\boldsymbol{x}$, we use $n(i)$ to denote the underlying bus index.[1] Specifically, let

$$D_{ij} = \begin{cases} A_{ij} & \text{if nodes } n(i) \text{ and } n(j) \\ & \text{belong to the same control area} \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

$$E_{ij} = \begin{cases} A_{ij} & \text{if nodes } n(i) \text{ and } n(j) \\ & \text{belong to different control areas} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

yielding $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{E}$. The key is to identify matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ such that $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ and the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$. To ensure convergence, we introduce a diagonal matrix $\bar{\boldsymbol{E}}$ whose ith diagonal entry equals

$$\bar{E}_{ii} \stackrel{\text{def}}{=} \sum_{j \neq i} |A_{ij}|. \tag{12}$$

We propose the following matrix-splitting design

$$\boldsymbol{M} = \boldsymbol{D} + \alpha \bar{\boldsymbol{E}} \tag{13}$$

$$\boldsymbol{N} = \alpha \bar{\boldsymbol{E}} - \boldsymbol{E}, \tag{14}$$

where α is a scalar parameter. In the Appendix Proposition 1, it is shown for $\alpha \geq 1/2$, the matrix-splitting iterative scheme in (9) converges to the centralized solution using the splitting designed in (13), (14). The centralized solution is given by directly, as opposed to iteratively, solving system (6).

C. Proposed Algorithm and Analysis of Information Communication Requirements

Besides our choice of matrix splitting satisfying conditions necessary for convergence, we also note that the only matrix to invert, namely $(\boldsymbol{D} + \alpha\bar{\boldsymbol{E}})$, is block diagonal and therefore can be inverted locally within each area without any communication between neighboring control areas since the inverse of a block diagonal matrix remains block-diagonal.

We address the question of which information needs to be communicated for each control area to calculate its local Gauss-Newton update, $\Delta \boldsymbol{x}^t = [\Delta \boldsymbol{x}_1^t \Delta \boldsymbol{x}_2^t \ldots \Delta \boldsymbol{x}_N^t]$, where there are a total of N control areas. Similarly, we can partition $\boldsymbol{b}$ from (8) into different components corresponding to each control area as $\boldsymbol{b} = [\boldsymbol{b}_1 \ldots \boldsymbol{b}_N]$. To illustrate the local computation required for the Gauss-Newton update, consider a network with two control areas. Then, the matrix-splitting iterative updates in (9) become

$$\begin{bmatrix} \Delta \boldsymbol{x}_1^{(t+1)} \\ \Delta \boldsymbol{x}_2^{(t+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_1^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_2^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_1^{(t)} \\ \Delta \boldsymbol{x}_2^{(t)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{M}_1^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_2^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} \tag{15}$$

In general, control area J has Gauss-Newton step update given by

$$\Delta \boldsymbol{x}_J^{t+1} = \boldsymbol{M}_J^{-1} \left[\sum_{L=1}^{N} \boldsymbol{N}_{JL} \Delta \boldsymbol{x}_L^t + \boldsymbol{b}_J \right]. \tag{16}$$

We already saw that $\boldsymbol{M}_J^{-1}$ can be calculated independently in each control area. We consider what information needs to be communicated in order to calculate the matrices $\{\boldsymbol{N}_{JL}\}$ and $\boldsymbol{b}_J$. Let the set of border buses in control area J be denoted Ω_J. These are the buses with a neighbor in another control area.

[1]For example, in a two node system, $\boldsymbol{x}^T = [\theta_1 \; V_1 \theta_2 \; V_2]$, and

$$n(i) = \begin{cases} 1 & \text{if } i = 1, 2 \\ 2 & \text{if } i = 3, 4. \end{cases}$$

Algorithm 1: Distributed Gauss-Newton Method for State Estimation

In parallel, each control area $1 \leq J \leq N$ does:
Initialization (Flat Start): $\boldsymbol{x}_J^{(0)} = \{\boldsymbol{\theta}_J^{(0)}, \boldsymbol{V}_J^{(0)}\}$ with $\boldsymbol{\theta}_J^{(0)} = \mathbf{0}$ and $\boldsymbol{V}_J^{(0)} = \mathbf{1}$.
for $k := 0$ *to* K **do**
  $\Delta \boldsymbol{x}_J^{(0,k)} = \mathbf{0}$
  for $t := 0$ *to* T **do**
    Calculate $\Delta \boldsymbol{x}_J^{(t+1,k)}$ from (16).
    for $a \in \Omega_J$ **do**
      Send $\Delta \boldsymbol{x}_a^{(t+1,k)}$ to neighboring nodes in other control areas.
      if $\widehat{P}_a$ *or* $\widehat{Q}_a \in \mathcal{Z}$ *and finished receiving neighbors updates* **then**
        Send $\{\Delta \boldsymbol{x}_b^{(t+1,k)}\}$ to neighboring nodes $b \in \mathcal{N}_a$ in other control areas.
      end
    end
  end
  $\boldsymbol{x}_J^{(k+1)} = \boldsymbol{x}_J^k - \Delta \boldsymbol{x}_J^{(k,T)}$
  for $a \in \Omega_J$ **do**
    Send $\boldsymbol{x}_a^{(k+1)}$ to neighbors $b \in \mathcal{N}_a$ in other control areas.
    if $\widehat{P}_a$ *or* $\widehat{Q}_a \in \mathcal{Z}$ *and finished receiving neighbors estimates* **then**
      Send $\{\boldsymbol{x}_b^{(k+1)}\}$ to neighboring nodes $b \in \mathcal{N}_a$.
    end
  end
end

Due to the locality of the measurement functions (whose functional form is included in Appendix (20)–(25) for reference), the matrix $\boldsymbol{N}_{JL}$ will be zero unless control areas J and L are neighboring (i.e., there is a transmission line connecting a bus in J to a bus in L) and otherwise sparse. Non-zero entries of $\boldsymbol{N}_{JL}$ can be attributed to measurements of power flow along transmission lines connecting areas J and L and to measurements of power injections at border buses. Similarily, such measurements are the only non-zero contributions to $\boldsymbol{b}_J$ from other control areas. The sparseness of the $\boldsymbol{N}_{JL}$ matrices allows for a limited communication of information between neighboring control areas. We emphasize that the information needed from other areas to calculate the matrices $\{\boldsymbol{N}_{JL}\}$ and $\boldsymbol{b}_J$ is communicated only once per Gauss-Newton iteration.

The communication exchange is detailed in Algorithm 1. It is important to stress that the centralized Gauss-Newton method consists only of the outer loop of Gauss-Newton iterations in (3) which produces a sequence of estimates using the exact Gauss-Newton step obtained from directly solving (6). We refer to these estimates as the centralized estimates, $\{\boldsymbol{x}_*^{(k)}\}$. In contrast, our distributed method iteratively solves (6) using the matrix-splitting scheme (i.e. inner loop) in (9). Let $\Delta\widetilde{\boldsymbol{x}}^{(t,k)}$ be the approximation to the Gauss-Newton step at inner-loop iteration t and outer-loop iteration k, and let the sequence of iterates produced by the distributed method using T inner-loop iterations be given by $\widetilde{\boldsymbol{x}}^{(k+1)} = \widetilde{\boldsymbol{x}}^{(k)} - \Delta\widetilde{\boldsymbol{x}}^{(T,k)}$.

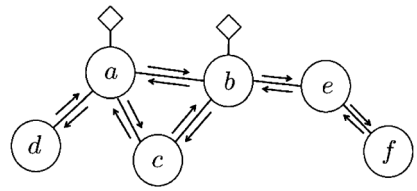


Fig. 3. Example to illustrate information exchange of Algorithm 1. There are power injection measurements at nodes a and b, signified by a diamond-shaped sensor. Communication occurs directly between neighbors.

We summarize the communication requirements of our algorithm. For clarity of exposition, consider each bus to be a separate control area. Communication between buses in the same control area is considered negligible compared to communication between buses in neighboring control areas. If there are no power injection measurements present at node a, then node a needs to communicate to its neighbors only its current state estimate $\widetilde{\boldsymbol{x}}_a^{(k)}$ at each outer-loop iteration and its Gauss-Newton step estimate $\Delta\widetilde{\boldsymbol{x}}_a^{(t,k)}$ at each inner-loop iteration. We illustrate the additional information exchange for nodes with a power injection measurement using the simple example in Fig. 3.

Let power injections be measured at nodes a and b. At initialization of the Gauss-Newton method,

- Node a sends the value of its real and reactive power injection measurements, $\widehat{P}_a$ and $\widehat{Q}_a$, to its neighbors b, c, and d.
- Node a receives $\widehat{P}_b$ and $\widehat{Q}_b$ from node b.

At outer-loop iteration k,

- Node a sends $\{\widetilde{\boldsymbol{x}}_a^{(k)}, \widetilde{\boldsymbol{x}}_b^{(k)}, \widetilde{\boldsymbol{x}}_c^{(k)}, \widetilde{\boldsymbol{x}}_d^{(k)}\}$ to its neighbors b, c, and d.
- Node a receives $\{\widetilde{\boldsymbol{x}}_b^{(k)}, \widetilde{\boldsymbol{x}}_c^{(k)}, \widetilde{\boldsymbol{x}}_e^{(k)}\}$ from node b. Node a also receives $\widetilde{\boldsymbol{x}}_c^{(k)}$ from node c and $\widetilde{\boldsymbol{x}}_d^{(k)}$ from node d.

At inner-loop iteration t,

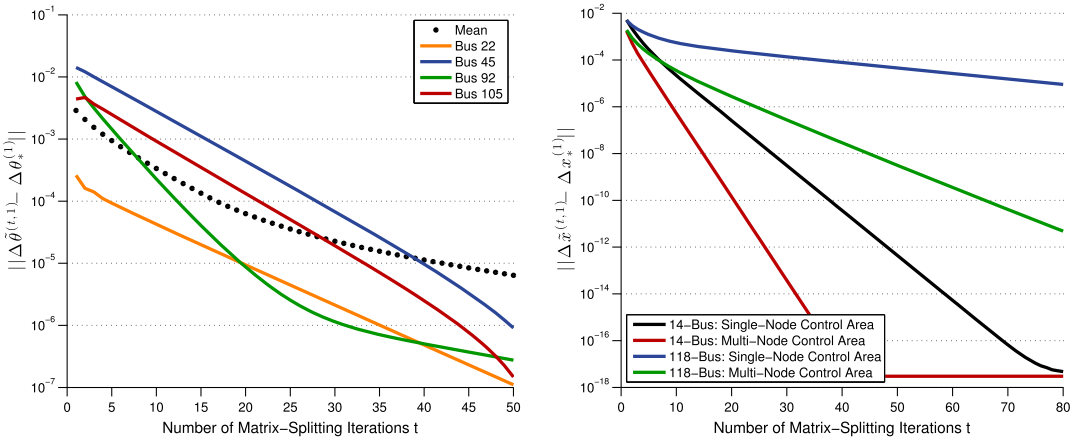
- Node a sends $\{\Delta\widetilde{\boldsymbol{x}}_a^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_b^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_d^{(t,k)}\}$ to its neighbors b, c, and d.
- Node a receives $\{\Delta\widetilde{\boldsymbol{x}}_b^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}, \Delta\widetilde{\boldsymbol{x}}_e^{(t,k)}\}$ from node b. Node a also receives $\Delta\widetilde{\boldsymbol{x}}_c^{(t,k)}$ from node c and $\Delta\widetilde{\boldsymbol{x}}_d^{(t,k)}$ from node d.

The distributed algorithm uses a finite number of matrix-splitting iterations, T, to calculate the Gauss-Newton step. Due to truncation error, this will not be exactly equal to the Gauss-Newton step, $\Delta\boldsymbol{x}^{(k)}$, given by directly solving (6). Therefore, we must distinguish between the centralized estimates, $\{\boldsymbol{x}_*^{(k)}\}$, and the distributed estimates, $\{\widetilde{\boldsymbol{x}}^{(k)}\}$. Let $\boldsymbol{y}_*^{(k)}$ be the solution of the linear system

$$\boldsymbol{J}\left(\widetilde{\boldsymbol{x}}^{(k)}\right)^T \boldsymbol{W}\boldsymbol{J}\left(\widetilde{\boldsymbol{x}}^{(k)}\right)\boldsymbol{y}_*^{(k)} = -\boldsymbol{J}\left(\widetilde{\boldsymbol{x}}^{(k)}\right)^T \boldsymbol{W}\left(\boldsymbol{z} - \boldsymbol{h}\left(\widetilde{\boldsymbol{x}}^{(k)}\right)\right) \tag{17}$$

The iterative scheme in (9) forms a discrete linear dynamic system and therefore exponentially converges to the solution $\boldsymbol{y}_*^{(k)}$ according to [34, Theorem 6.1], provided $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$ (see Appendix Proposition 1). Formally, we have the following theorem about the convergence:

Theorem 1: The distributed Newton step calculation as described in (9) exponentially converges to the solution of the linear system in (17). The convergence speed is determined by $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$.



(a) 118-bus system. (b) Different communication schemes.

Fig. 4. These figures show the exponential convergence of the distributed Gauss-Newton step. In (a), the convergence is shown for the Gauss-Newton step update of the phase angles at individual buses. In (b), the convergence of the distributed Gauss-Newton step estimates to the centralized exact Gauss-Newton step is faster in the ("multi-node control area") setting than in the ("single-node control area") setting.

IV. Numerical Results

We present our numerical results in three sections. First, we study the convergence properties of the matrix-splitting (i.e., inner-loop iterations). Since we are using an iterative method to solve for the Gauss-Newton step, it is important to see how this approximation numerically compares to directly solving for the Gauss-Newton step. Second, we demonstrate the performance of our algorithm in various settings, including a large-scale realistic system with 1,354 buses. Third, we present results towards designing an optimal splitting with respect to the parameter α and to the control area partitioning. Interesting open questions remain in this area.

A. Convergence of Matrix-Splitting Iterations

To study the convergence of the distributed algorithm with respect to the matrix-splitting (i.e., inner-loop) iterations, we calculate at each matrix-splitting iteration t, the error on the distributed Newton step $\|\Delta\widetilde{\boldsymbol{x}}^{(t,k)} - \Delta\boldsymbol{x}_*^{(k)}\|$. Fig. 4 demonstrates the exponential convergence of the distributed Gauss-Newton step to the centralized Gauss-Newton step for $k = 1$. The case studies include the IEEE 14-bus and IEEE 118-bus systems. Additionally, we study the convergence under two different communication schemes. In the ("Single-Node Control Area") scheme, each node is considered its own control area. The single-node control area setting is a natural limit for understanding the behavior of the algorithm in terms of the size of the control areas and the degree to which the calculation is distributed. In the ("Multi-Node Control Area") setting, several nodes are grouped into a single control area. From Fig. 4(b), the convergence of the ("Multi-Node Control Area") setting is faster than the ("Single-Node Control Area") setting. However, each matrix-splitting iteration is computationally more costly for the ("Multi-Node Control Area") setting, so there is a tradeoff between required number of iterations and the computational cost of each iteration.

We use the partitioning of the 14-bus system into the four control areas given in Fig. 1 [14]. For the 118-bus system, we use the nine control area partitioning from [12]. For the 14-bus system, the measurement configuration used is shown in Fig. 1. The types and locations of measurements used for the 118-bus system tests are given in Table I. The measurements are perturbed by additive Gaussian noise where the variance of the measurements used was taken from [12]. The inverse covariance matrix $\boldsymbol{R}$ is used as the weighting matrix $\boldsymbol{W}$.

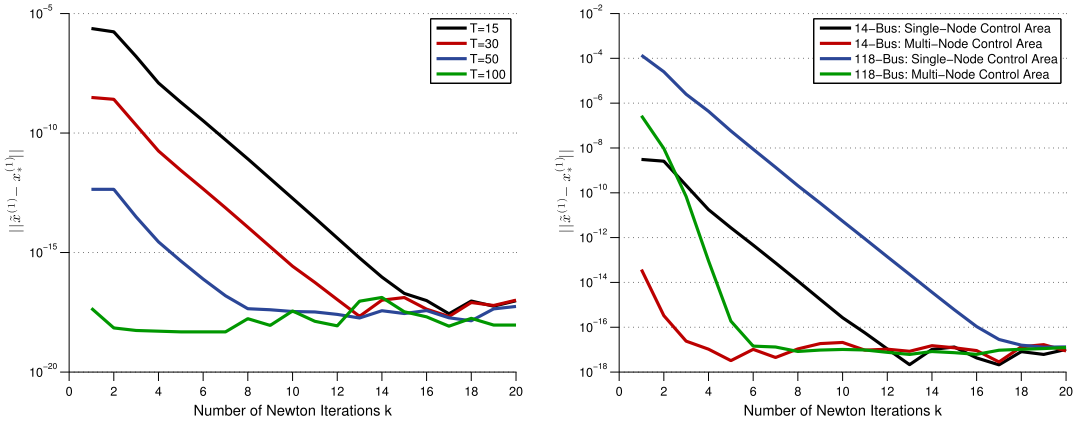
TABLE I
118-Bus Measurement Configuration

Measurement Type	Locations (Bus Index)
Missing Power Flows	(2,12), (9,10), (24,72), (93,94)
Power Injections	1, 2, 3, 5, 18, 19, 21, 22, 25, 33, 34, 35, 36, 55, 56, 69, 71, 72, 77, 80, 81, 86, 88, 90, 98, 106,107 110, 114, 118
Missing voltage phase angles	3, 5, 14, 15, 18, 19, 20, 21, 26, 33, 38, 39, 41, 44, 45, 51, 63, 64, 67, 68, 51, 63, 64, 67, 68, 63, 64, 67, 68, 69, 73, 74, 75, 76, 79, 82, 84, 89, 92, 94, 96, 97, 99, 100, 101, 105, 106, 107

As another measure of the convergence of the matrix-splitting iterations, it is interesting to examine the difference between the centralized and distributed estimates as a function of outer-loop iterations. Fig. 5 shows the error of the distributed estimates with respect to the centralized estimates. The linear system in (6) is identical for the distributed and centralized approach only at the first Gauss-Newton (i.e., outer-loop) iteration. This is due to truncation error resulting from using only a finite number, T, of matrix-splitting iterations. Despite propagating an inexact Gauss-Newton step at each Gauss-Newton iteration, the distributed estimates agree with the centralized estimates within several tens of iterations.

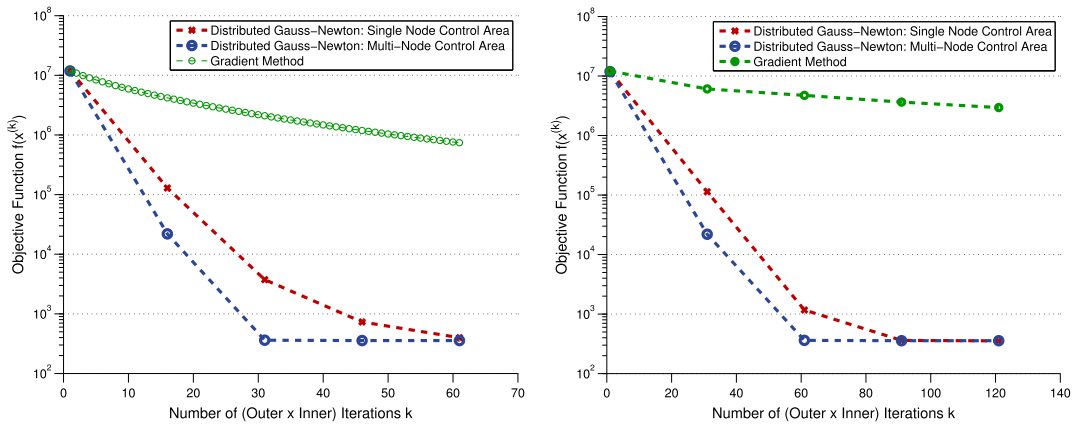
B. Performance

In Fig. 6, we show the convergence of $f(\boldsymbol{x}^{(k)})$ as a function of the total number of iterations (number of inner-loop iterations times number of outer-loop iterations) for the distributed Gauss-Newton method and gradient method. In Fig. 6(a), we use the gradient method with a constant step size, which can be implemented in a fully distributed way. The gradient method with



(a) 14-bus system. (b) Using $T = 30$.

Fig. 5. These figures show the difference between the distributed and centralized estimates as a function of the number of Gauss-Newton iterations. In (a), we study this for different values of T, the number of matrix-splitting iterations, for the 14-bus system. We see that for $T = 100$, the distributed and exact Gauss-Newton step agree up to machine precision for all iterations. In (b), we use $T = 30$ iterations and compare for different network sizes and communication schemes.



(a) Gradient method with constant stepsize. (b) Gradient method with backtracking step-size.

Fig. 6. The convergence of the objective function for the 118-bus system is shown with respect to the total number of iterations. In a), the gradient method uses a constant stepsize (i.e., no inner iterations), and the distributed Gauss-Newton method uses $T = 15$ inner iterations per outer iteration. In b), the gradient method uses a backtracking line search. This requires 30 inner iterations. To compare, we use $T = 30$ matrix-splitting iterations in b). Using a backtracking line search, the gradient method is no longer a distributed method.

constant step size requires no inner iterations. In Fig. 6(b), we compare the gradient method using a backtracking line search to determine the step-size. We note that the backtracking line search requires central coordination. These plots show the advantage of Newton-type methods over gradient descent with respect to rate of convergence.

Next, we test the performance of our algorithm on a larger, more realistic system. The algorithm is implemented in C++ using MPI [36] to run over multiple processors. A separate process is assigned to each control area, and using the MPI graph communicator functionality, we design the communication so that two control areas can communicate only if there is a tie-line connecting them. Our results were obtained using the Blue Waters Supercomputing System [37]. In Fig. 7(a), we show the convergence of the objective function in terms of runtime in seconds for different control area partitionings of the PEGASE 1,354-bus test system [35], [38]. The centralized scheme (using a single control area) is notably slower. Increasing the number of control areas and hence degree of distributivity decreases the runtime. The breakdown of CPU time spent on computation and communication for a single distributed Gauss-Newton iteration is provided in Fig. 7(b).

For the measurement configuration, we randomly place power injection measurements and PMUs at 10% of the buses and 33% of the buses, respectively. Power flow measurements are taken at 90% of the transmission lines. To partition the 1,354-bus network into control areas as a pre-processing step, we use a routine written by J. Hespanha [39], which clusters based on spectral factorization.

Additionally, we study the robustness of our algorithm to communication failures between control areas in Fig. 8. The probability of failure p_f is the probability that a communication failure between two control areas will occur. We see that our algorithm is fairly robust under such communication dropouts and demonstrates a decrease in the rate of convergence as the probability of communication failure increases.

C. Towards an Optimal Matrix Splitting

Recall from Theorem 1 that the convergence of the matrix-splitting iterations is determined by the spectral radius

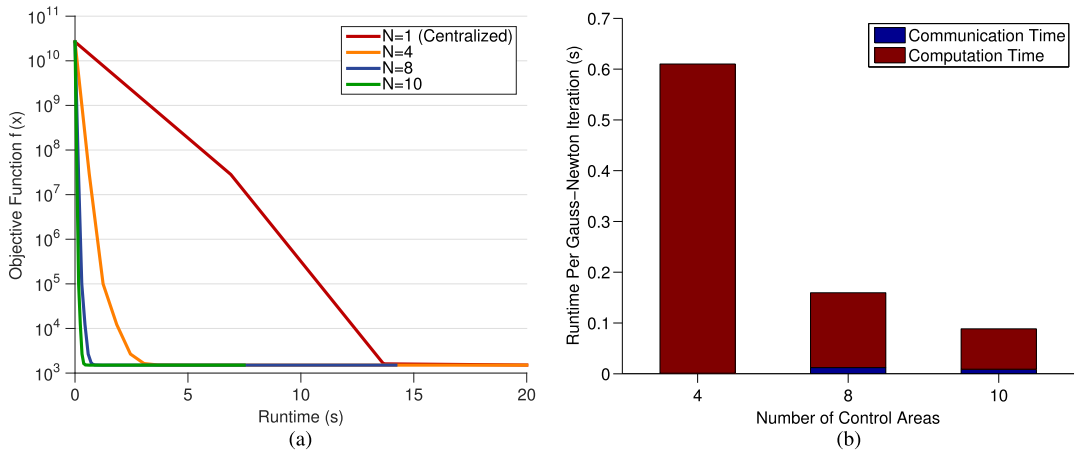


Fig. 7. For a system with 1,354 buses, in a) the convergence of the objective function $f(\boldsymbol{x})$ in terms of runtime in seconds is shown for different control area partitionings. With more control areas (higher degree of distributivity), the runtime needed to converge is smaller. In b), the breakdown of time spent in a single outer-loop iteration for computation versus communication is shown. As the number of control areas grows, communication time remains constant as desired, while computation decreases since the problem solved by each processor is smaller.

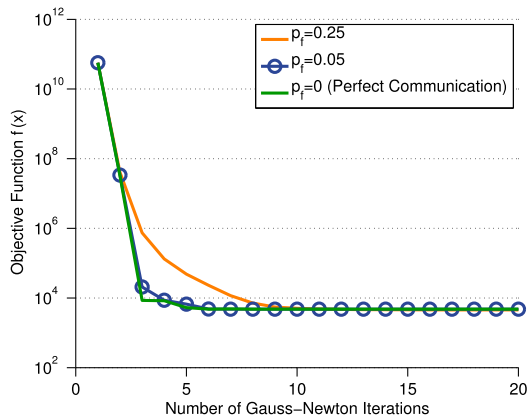


Fig. 8. Test performance of the 1,354-bus system in presence of communication failure between control areas.

$\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$. The value of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ depends on the way the network is partitioned into control areas and the value of the splitting parameter α from (13), (14). Optimizing the convergence rate with respect to the partitioning is an interesting open question. Here we present some initial numerical results. Fig. 9(a) shows the dependence of the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ on the parameter α and the number of control areas used to partition the network. There are a combinatorial number of ways to partition a network of n nodes into N control areas. For each value of N, the values shown in Fig. 9 are averaged over all possible integer compositions of n into N parts. For $N = 5$, the control area partitioning achieving the minimal spectral radius is shown in Fig. 9(b). We see that, for each possible partitioning, the spectral radius is minimal at $\alpha = 1/2$. As stated in Section III-B, for $\alpha \geq 1/2$, we have $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$. We note that this is a sufficient and not necessary condition on α, so one might want to try to tune α accordingly. Analytical results relating $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ to α in a simplified setting are available in the Appendix Proposition 2.

V. Conclusion

In this paper, we use matrix-splitting techniques to develop a new fully distributed Gauss-Newton algorithm for power system state estimation. We consider the non-linear AC measurement model and a multi-area setting. The method incorporates both traditional SCADA measurements, as well as PMU measurements. The algorithm requires limited sharing of information between neighboring control areas and allows for each control area to store and compute only its local estimate. Numerical experiments verify the convergence properties of this algorithm and the advantage of Newton-type methods for state estimation. For future work, it will be interesting to investigate how to incorporate distributed bad data detection and network observability into our algorithm.

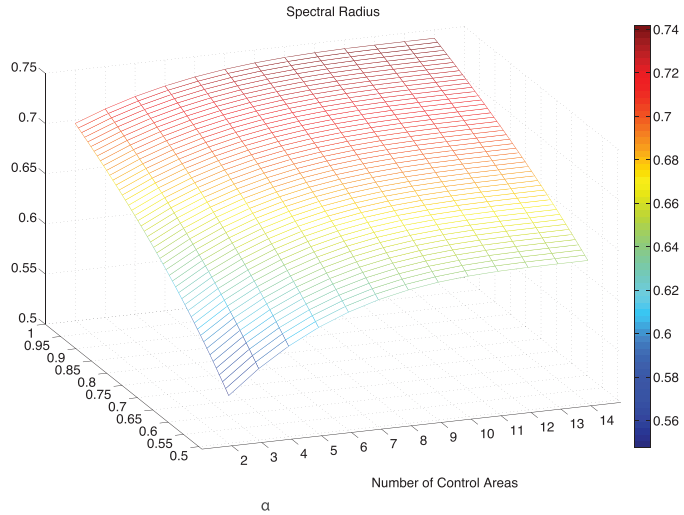
Appendix

A. Proof of Matrix-Splitting Convergence

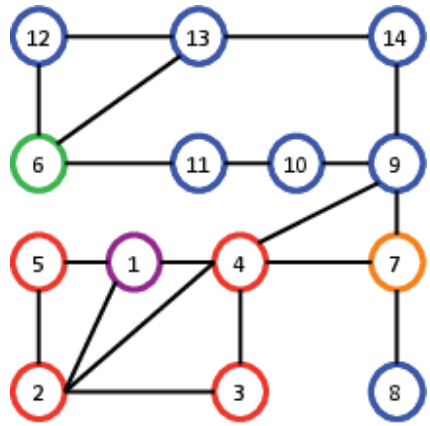
First, we note the following statements. The sequence $\{\Delta\boldsymbol{x}^t\}$ in (9) converges to its limit $\Delta\boldsymbol{x}^*$ as $t \to \infty$ if and only if the spectral radius of the matrix $\boldsymbol{M}^{-1}\boldsymbol{N}$ is strictly less than 1 [25]. Furthermore, if the sequence converges, the limit $\Delta\boldsymbol{x}^*$ is the solution of the system, (i.e., $\boldsymbol{A}\Delta\boldsymbol{x}^* = \boldsymbol{b}$). In order to have the spectral radius $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$, it is sufficient to have $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N} \succ \boldsymbol{0}$ and $\boldsymbol{M} + \boldsymbol{N} \succ \boldsymbol{0}$ [40]. We have the following proposition which ensures convergence of the matrix-splitting iterates in (9).

Proposition 1: Given a positive definite matrix $\boldsymbol{A}$, let $\boldsymbol{M} = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{N} = \alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ where $\boldsymbol{D}$, $\boldsymbol{E}$, and $\bar{\boldsymbol{E}}$ are given in (10)–(12). Then, for $\alpha \geq 1/2$, $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$.

a) Proof: By Theorem 2.5.3 of [40], to prove that $\rho(\boldsymbol{M}^{-1}\boldsymbol{N}) < 1$, it is sufficient to show that $\boldsymbol{M} - \boldsymbol{N}$ and $\boldsymbol{M} + \boldsymbol{N}$ are both positive definite. Using the Gauss-Newton method, the formula for $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ in (7) is positive definite by construction. Thus, it is sufficient to show that $\boldsymbol{M} + \boldsymbol{N} = \boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$



(a)



(b)

Fig. 9. (a) Dependence of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ on α and control area partitioning for the 14-bus test case. Using $N = 14$ control areas, the algorithm is fully-distributed. Using a single control area ($N = 1$), the algorithm is centralized. This plot exhibits the tradeoff between the degree to which the calculation is distributed and the rate of convergence. In addition, the minimal values of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ are achieved at $\alpha = 1/2$. (b) Configuration achieving minimal value of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ with $N = 5$ control areas is shown. It is interesting to note that the optimal communication network is not necessarily the same as the underlying power grid structure as evidenced by the disconnection of node 8 from the rest of its control area.

is positive definite. As a corollary to the Gershgorin Circle Theorem [25], we only need to show that $\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ is strictly diagonally dominant with strictly positive diagonal entries.[2] Given that $\alpha \geq 1/2$,

$$[\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}]_{ii} = D_{ii} + 2\alpha\bar{E}_{ii}$$
$$> 2\alpha \sum_{j\neq i} |A_{ij}| \quad (18a)$$
$$\geq \sum_{j\neq i} \left|[\boldsymbol{D} + 2\alpha\bar{\boldsymbol{E}} - \boldsymbol{E}]_{ij}\right|, \quad (18b)$$

where (18a) follows from the definition of $\bar{\boldsymbol{E}}$ and from the fact that $D_{ii} > 0$. Since $\boldsymbol{A}$ is assumed to be positive definite, its diagonal entries must be strictly positive. The inequality in (18b) follows from (18a) since by construction the support of $\boldsymbol{D}$ and $\boldsymbol{E}$ are complimentary, meaning that for all i, j if D_{ij} is nonzero then $E_{ij} = 0$ and vice versa. Furthermore, the contribution from $2\alpha\bar{\boldsymbol{E}}$ in (18b) is zero since the off-diagonal terms of $\bar{\boldsymbol{E}}$ are zero. ■

See (19) at the bottom of the page.

[2]A matrix $\boldsymbol{A}$ is strictly diagonal dominant if $|A_{ii}| > \sum_{j\neq i} |A_{ij}|$ for all i.

B. Proof of Communication Requirements

The local nature of the measurement equations is key for allowing our algorithm to have limited communication requirements. We introduce the following notations to denote different kinds of measurement equations and emphasize which states that they depend upon. Let the neighbor set of node a be denoted $\mathcal{N}_a$. The exact mathematical form of the non-linear measurement model equations can be found in [1].

1) If $z_k = \widehat{P}_{ab}$, then

$$h_k(\boldsymbol{x}) = h_{P(a,b)}(\theta_a, V_a, \theta_b, V_b). \quad (20)$$

$$\begin{aligned} y_i^{(t)} = \alpha &\left(\sum_{\{j|n(j)\in\mathcal{N}_a^*\}} |A_{ij}| \right) \Delta x_i^{(t)} - \sum_{\{j|n(j)\in\mathcal{N}_a^*\}} E_{ij}\Delta x_j^{(t)} - 2\left\{\sigma_{a,\theta}(\widehat{\theta}_a - \theta_a) + \sigma_{a,V}(\widehat{V}_a - V_a)\right. \\ &+ \sum_{\{j|b\equiv n(j)\in\mathcal{N}_a\}} \left[\sigma_{P(a,b)} \frac{\partial h_{P(a,b)}}{\partial x_i} \left(\widehat{P}_{ab} - h_{P(a,b)}(\boldsymbol{x}_a, \boldsymbol{x}_b)\right) + \sigma_{Q(a,b)} \frac{\partial h_{Q(a,b)}}{\partial x_i} \left(\widehat{Q}_{ab} - h_{Q(a,b)}(\boldsymbol{x}_a, \boldsymbol{x}_b)\right)\right. \\ &\left. + \sigma_{b,P} \frac{\partial h_{b,P}}{\partial x_i} \left(\widehat{P}_b - h_{b,P}\left(\boldsymbol{x}_b, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_b}\right)\right) + \sigma_{b,Q} \frac{\partial h_{b,Q}}{\partial x_i} \left(\widehat{Q}_b - h_{b,Q}\left(\boldsymbol{x}_b, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_b}\right)\right)\right] \\ &\left. + \sigma_{a,P} \frac{\partial h_{a,P}}{\partial x_i} \left(\widehat{P}_a - h_{a,P}\left(\boldsymbol{x}_a, \{\boldsymbol{x}_c\}_{c\in\mathcal{N}_a}\right)\right) + \sigma_{a,Q} \frac{\partial h_{a,Q}}{\partial x_i} \left(\widehat{Q}_a - h_{a,Q}\left(\boldsymbol{x}_a, \{\boldsymbol{x}_c\}_c \in \mathcal{N}_a\right)\right)\right\} \end{aligned} \quad (19)$$

2) If $z_k = \widehat{Q}_{ab}$, then

$$h_k(\boldsymbol{x}) = h_{Q(a,b)}(\theta_a, V_a, \theta_b, V_b). \tag{21}$$

3) If $z_k = \widehat{P}_a$, then

$$h_k(\boldsymbol{x}) = h_{a,P}\left(\theta_a, V_a, \{\theta_b\}_{b\in\mathcal{N}_a}, \{V_b\}_{b\in\mathcal{N}_a}\right). \tag{22}$$

4) If $z_k = \widehat{Q}_a$, then

$$h_k(\boldsymbol{x}) = h_{a,Q}\left(\theta_a, V_a, \{\theta_b\}_{b\in\mathcal{N}_a}, \{V_b\}_{b\in\mathcal{N}_a}\right). \tag{23}$$

5) If $z_k = \widehat{\theta}_a$, then

$$h_k(\boldsymbol{x}) = h_{\theta,a}(\theta_a). \tag{24}$$

6) If $z_k = \widehat{V}_a$, then

$$h_k(\boldsymbol{x}) = h_{V,a}(V_a). \tag{25}$$

Applying the iterative scheme of (9) to the state estimation problem with our choice of matrix splitting, we obtain

$$\Delta\boldsymbol{x}^{(t+1)} = \left(D + \alpha\bar{E}\right)^{-1}\left[(\alpha\bar{E} - E)\Delta\boldsymbol{x}^{(t)} + \nabla f(\boldsymbol{x})\right]. \tag{26}$$

We denote the term within the bracket as:

$$\boldsymbol{y}^{(t)} \stackrel{\text{def}}{=} (\boldsymbol{\alpha\bar{E}} - \boldsymbol{E})\Delta\boldsymbol{x}^{(t)} + \nabla f(\boldsymbol{x}). \tag{27}$$

In terms of the N control areas, (26) can be written in block-form as

$$\begin{bmatrix} \Delta\boldsymbol{x}_1^{(t+1)} \\ \vdots \\ \Delta\boldsymbol{x}_N^{(t+1)} \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{M}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{M}}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1^{(t)} \\ \vdots \\ \boldsymbol{y}_N^{(t)} \end{bmatrix}, \tag{28}$$

where $\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_N$ are the diagonal blocks of $(\boldsymbol{D} + \alpha\bar{\boldsymbol{E}})^{-1}$ corresponding to the different control areas. In the case that each bus is treated as a control area, the matrix inversion is reduced to the inversion of a scalar number. This simplified setting was treated in our previous work [41].

We utilize the particular structure of the power grid state estimation problem in order to verify the information from neighboring areas needed to calculate the entries of $\Delta\boldsymbol{x}^{(t)}$. We introduce the following quantities:

$$\sigma_{P(a,b)} = \begin{cases} w_k & z_k = \widehat{P}_{ab} \\ 0 & \widehat{P}_{ab} \notin \mathcal{Z} \end{cases}, \sigma_{Q(a,b)} = \begin{cases} w_k & z_k = \widehat{Q}_{ab} \\ 0 & \widehat{Q}_{ab} \notin \mathcal{Z} \end{cases},$$
$$\sigma_{a,P} = \begin{cases} w_k & z_k = \widehat{P}_a \\ 0 & \widehat{P}_a \notin \mathcal{Z} \end{cases}, \sigma_{a,Q} = \begin{cases} w_k & z_k = \widehat{Q}_a \\ 0 & \widehat{Q}_a \notin \mathcal{Z} \end{cases},$$
$$\sigma_{a,\theta} = \begin{cases} w_k & z_k = \widehat{\theta}_a \\ 0 & \widehat{\theta}_a \notin \mathcal{Z} \end{cases}, \sigma_{a,V} = \begin{cases} w_k & z_k = \widehat{V}_a \\ 0 & \widehat{V}_a \notin \mathcal{Z} \end{cases},$$

where w_k are the diagonal entries of the weighting matrix $\boldsymbol{W}$ (see Assumption 1). In addition, we define the set of "2-hop" neighbor nodes as nodes which are not direct neighbors ("1-hop") but share a common neighbor, $\mathcal{N}_a^{\dagger} \stackrel{\text{def}}{=} \{b | b \notin \mathcal{N}_a, \mathcal{N}_a \cap \mathcal{N}_b \neq \emptyset\}$. Let $\mathcal{N}_a^* \stackrel{\text{def}}{=} \mathcal{N}_a \cup \mathcal{N}_a^{\dagger}$ be the union of the sets of "1-hop" and "2-hop" neighbors.

Lemma 1: Let $a := n(i)$ be the node index corresponding to entry i of $\boldsymbol{y}^{(t)}$, and let the set $\boldsymbol{x}_a = (\theta_a, V_a)$ refer to the set of estimates of both the voltage angle and magnitude at bus a at the current Gauss-Newton iteration. The components of the updates to $\boldsymbol{y}^{(t)}$ from (27) can be reduced to the form given in (19) for a single bus.

a) Proof: We determine the sparsity pattern of $\boldsymbol{A}$ as a function of the network connectivity. From the AC measurement model equations, we have that

$$\left[\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{W}\boldsymbol{J}(\boldsymbol{x})\right]_{ij} = \sum_{k=1}^{m} w_k J_{ki}(\boldsymbol{x}) J_{kj}(\boldsymbol{x}) \tag{29}$$

is nonzero if and only $n(j) \in \mathcal{N}_a^*$. The entries of (27) simplify to the expression in (19). We note that if power injection measurements are not present, the sparsity is reduced to A_{ij} being nonzero if and only if $n(j) \in \mathcal{N}_a$. ∎

Furthermore, from the proof of Lemma 1, node a in control area J only needs information from nodes in $\mathcal{N}_a^*$ to calculate entries of $\widehat{\boldsymbol{M}}_J$. This specifies the subset of external information needed to calculate the updates to $\Delta\boldsymbol{x}^{(t)}$ in (28).

C. Proof of Some Properties of the Spectral Radius

As a starting point towards an analytical analysis of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$, assume there are only power flow and PMU measurements available, and we consider the "DC" linear measurement model, $\boldsymbol{z} = \boldsymbol{H\theta}$. For additional details of our distributed state estimation algorithm using the DC approximation, we refer readers to [41]. In this setting, we can show that $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is

1) an increasing function of α and
2) a decreasing function of the number of diagonal blocks in the splitting of $\boldsymbol{A}$.

For the DC model, we have that $\boldsymbol{A} = \boldsymbol{H^T W} H$. We assume that $\boldsymbol{A}$ is positive definite. In fact, it is sufficient to assume that $\boldsymbol{H}$ has full column rank. Without power injection measurements, the entries of $\boldsymbol{A}$ are given by

$$A_{ij} = \begin{cases} \sigma_{i,\theta} + \sum_{l\in\mathcal{N}_i}\left(\sigma_{P(i,l)} + \sigma_{P(l,i)}\right)B_{il}^2, & j = i \\ -\left(\sigma_{P(i,j)} + \sigma_{P(j,i)}\right)B_{ij}^2, & j \in \mathcal{N}_i \\ 0, & \text{otherwise,} \end{cases} \tag{30}$$

where B_{ij} is the susceptance of the transmission line connecting buses i and j.

From this we see that $\boldsymbol{A}$ has positive diagonal entries and non-positive off-diagonal entries. Furthermore, since $\boldsymbol{A}$ is symmetric and positive definite, $\boldsymbol{A}$ is a *Stieljes* matrix. Stieljes matrices have the property that their inverses are non-negative, $\boldsymbol{A}^{-1} \geq \boldsymbol{0}$, (Cor. 3.24) [42]. From Definition 3.28 in [42], for $n \times n$ real matrices, $\boldsymbol{A}, \boldsymbol{M}$, and $\boldsymbol{N}, \boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ is a regular splitting of the matrix $\boldsymbol{A}$ if $\boldsymbol{M}$ is nonsingular with $\boldsymbol{M}^{-1} \geq \boldsymbol{0}$ and $\boldsymbol{N} \geq \boldsymbol{0}$.

Lemma 2: The splitting in (13), (14) is a regular splitting for $\alpha \geq 0$.

a) Proof: Since $\boldsymbol{A}$ is a Stieljes matrix, $\boldsymbol{D}$ is also a Stieljes matrix. Upon adding a diagonal matrix with strictly positive entries to a Stieljes matrix, the matrix remains Stieljes. Therefore, for $\alpha \geq 0, \boldsymbol{M} = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ is a Stieljes matrix, and we have $\boldsymbol{M}^{-1} \geq \boldsymbol{0}$ and $\boldsymbol{N} \geq \boldsymbol{0}$. ∎

To show the desired properties of $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$, we utilize the following theorem from [42, Theorem 3.32].

Theorem 2 (Varga): Let $\boldsymbol{A} = \boldsymbol{M_1} - \boldsymbol{N_1} = \boldsymbol{M_2} - \boldsymbol{N_2}$ be two regular splittings of $\boldsymbol{A}$, where $\boldsymbol{A}^{-1} \geq \boldsymbol{0}$. If $\boldsymbol{0} \leq \boldsymbol{N_1} \leq \boldsymbol{N_2}$, then

$$0 \leq \rho\left(\boldsymbol{M_1}^{-1}\boldsymbol{N_1}\right) \leq \rho\left(\boldsymbol{M_2}^{-1}\boldsymbol{N_2}\right) < 1. \tag{31}$$

With this result, we have the following proposition that analytically characterizes the rate of convergence with respect to the matrix splitting parameter α and with respect to block size.

Proposition 2: In the DC setting and when no power injection measurements are present, $\rho(\boldsymbol{M}^{-1}\boldsymbol{N})$ is an increasing function of α for $\alpha \geq 0$. The rate of convergence of the iterative algorithm increases as block size increases or equivalently the number of blocks decreases.

b) Proof: Let $0 \leq \alpha < \alpha'$ and let $\boldsymbol{N_1} = \alpha\bar{\boldsymbol{E}} - \boldsymbol{E}$ and $\boldsymbol{N_2} = \alpha'\bar{\boldsymbol{E}} - \boldsymbol{E}$, where $\bar{\boldsymbol{E}}$ is defined as in (12). Then, $\boldsymbol{N_1} < \boldsymbol{N_2}$, so from Theorem 2 we have $\rho(\boldsymbol{M_1}^{-1}\boldsymbol{N_1}) \leq \rho(\boldsymbol{M_2}^{-1}\boldsymbol{N_2})$. This shows that the rate of convergence increases as α decreases. Consider a splitting with $\boldsymbol{M}_1 = \boldsymbol{D} + \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{M}_2 = \boldsymbol{D}' + \alpha\bar{\boldsymbol{E}}'$, where $\bar{\boldsymbol{E}}'$ is diagonal with entries given by $\bar{E}'_{ii} = \sum_{j=1}^{n} |A'_{ij}|$. Let $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A}_{11}, \ldots, \boldsymbol{A}_{ii}, \ldots, \boldsymbol{A}_{NN})$ and $\boldsymbol{D}' = \operatorname{diag}(\boldsymbol{A}_{11}, \ldots, \boldsymbol{A}'_{ii}, \ldots, \boldsymbol{A}_{NN})$, where

$$\boldsymbol{A}_{ii} = \begin{bmatrix} \boldsymbol{A}_{ii}^{(1,1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{ii}^{(2,2)} \end{bmatrix}, \tag{32}$$

and

$$\boldsymbol{A}'_{ii} = \begin{bmatrix} \boldsymbol{A}_{ii}^{(1,1)} & \boldsymbol{A}_{ii}^{(1,2)} \\ \boldsymbol{A}_{ii}^{(2,1)} & \boldsymbol{A}_{ii}^{(2,2)} \end{bmatrix}. \tag{33}$$

Then, we have $\boldsymbol{N}_1 = \boldsymbol{A} - \boldsymbol{D} - \alpha\bar{\boldsymbol{E}}$ and $\boldsymbol{N}_2 = \boldsymbol{A} - \boldsymbol{D}' - \alpha\bar{\boldsymbol{E}}'$. Since the off-diagonal entries of $\boldsymbol{A}$ are non-positive, by construction $\boldsymbol{A}_{ii} \leq \boldsymbol{A}'_{ii}$, so we have $\boldsymbol{N}_1 \leq \boldsymbol{N}_2$. From Theorem 2, we have $\rho(\boldsymbol{M_1}^{-1}\boldsymbol{N_1}) \leq \rho(\boldsymbol{M_2}^{-1}\boldsymbol{N_2})$. From (32) and (33), we see that the splitting $\boldsymbol{A} = \boldsymbol{M}_1 + \boldsymbol{N}_1$ results from dividing a given block into two smaller blocks. Since the selection of the block $\boldsymbol{A}'_{ii}$ was arbitrary, we see that decreasing the block size or equivalently increasing the number of blocks decreases the rate of convergence. ∎

This shows an interesting connection between the rate of convergence and the communication scheme, or degree to which the algorithm is distributed.

ACKNOWLEDGMENT

The authors would like to thank Ermin Wei at Northwestern University for her helpful discussions. We also thank the editor and reviewers for their constructive criticism and suggestions in improving the paper.

REFERENCES

[1] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC Press, 2004.
[2] A. Chakrabortty and P. Khargoneka, "Introduction to wide-area control of power systems," in *Proc. Amer. Control Conf.*, 2013, pp. 6773–6785.
[3] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 33–43, 2012.
[4] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, 2013.
[5] C.-H. Lin, "A distributed state estimator for electric power systems," *IEEE Trans. Power Syst.*, vol. 7, no. 2, pp. 551–557, May 1992.
[6] D. Falcao, F. Wu, and L. Murphy, "Parallel and distributed state estimation," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 724–730, May 1995.
[7] A. Gomez-Exposito, A. Abur, A. de la Villa Jaen, C. Gomez-Quiles, P. Rousseaux, and T. Van Cutsem, "A taxonomy of multi-area state estimation methods," *Elect. Power Syst. Res.*, vol. 81, pp. 1060–1069, 2011.
[8] G. Korres, A. Tzavellas, and E. Galinas, "A distributed implementation of multi-area power system state estimation on a cluster of computers," *Elect. Power Syst. Res.*, vol. 102, pp. 20–32, Sep. 2013.
[9] W. Jiang, V. Vittal, and G. Heydt, "A distributed state estimator utilizing synchronized phasor measurements," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 563–571, May 2007.
[10] W. Jiang, V. Vittal, and G. Heydt, "Diakoptic state estimation using phasor measurement units," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1580–1589, Nov. 2008.
[11] R. Ebrahimian and R. Baldick, "State estimation distributed processing," *IEEE Trans. Power Syst.*, vol. 15, no. 4, pp. 1240–1246, Nov. 2000.
[12] L. Zhao and A. Abur, "Multiarea state estimation using synchronized phasor measurements," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 611–617, 2005.
[13] A. G. Exposito, A. Abur, A. de la Villa Jaen, and C. Gomez-Quiles, "A multilevel state estimation paradigm for smart grids," *Proc. IEEE*, vol. 99, no. 6, pp. 952–976, Jun. 2011.
[14] G. N. Korres, "A distributed multiarea state estimation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 73–84, Feb. 2011.
[15] T. Van Cutsem, J. Horward, and M. Ribbens-Pavella, "A two-level static state estimator for electric power systems," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 8, pp. 3722–3732, Aug. 1981.
[16] L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1154–1169, 2012.
[17] X. Li and A. Scaglione, "Advances in decentralized state estimation for power systems," in *Proc. IEEE 5th Int. Workshop Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2013, pp. 428–431.
[18] Y. Hu, A. Kuh, T. Yang, and A. Kavcic, "A belief propagation based power distribution system state estimator," *IEEE Computat. Intell. Mag.*, vol. 6, no. 3, pp. 36–46, Aug. 2011.
[19] A. Conejo, S. De La Torre, and M. Canas, "An optimization approach to multiarea state estimation," *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 213–221, Feb. 2007.
[20] J. Carvalho and F. Barbosa, "Distributed processing in power system state estimation," in *Proc. 10th Mediterranean Electrotechnical Conf. (MELECON)*, May 2000, vol. 3, pp. 1128–1131.
[21] E. Caro, A. Conejo, and R. Minguez, "Decentralized state estimation and bad measurement identification: An efficient Lagrangian relaxation approach," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2500–2508, Nov. 2011.
[22] V. Kekatos and G. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.
[23] X. Tai, Z. Lin, M. Fu, and Y. Sun, "A new distributed state estimation technique for power networks," in *Proc. Amer. Control Conf.*, 2013, pp. 3338–3343.
[24] J. M. Ortega, *Matrix Theory: A Second Course*. New York, NY, USA: Springer, 1986.
[25] R. Varga, *Matrix Iterative Analysis*, ser. Springer Series in Computational Mathematics. New York, NY, USA: Springer, 2000.
[26] E. Wei, A. E. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization," in *Proc. Conf. Decision and Control*, 2010, pp. 1816–1821.
[27] S.-Y. Lin and C.-H. Lin, "An implementable distributed state estimator and distributed bad data processing schemes for electric power systems," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1277–1284, Aug. 1994.

[28] M. Shahidehpour and Y. Wang, *Communication and Control in Electric Power Systems: Applications of Parallel and Distributed Processing*. New York, NY, USA: Wiley-IEEE Press, 2003.
[29] T. S. Bi, X. H. Qin, and Q. X. Yang, "A novel hybrid state estimator for including synchronized phasor measurements," *Elect. Power Syst. Res.*, vol. 78, no. 8, pp. 1343–1352, Aug. 2008.
[30] S. Chakrabarti, E. Kyriakides, G. Ledwich, and A. Ghosh, "Inclusion of PMU current phasor measurements in a power system state estimator," *IET Gener., Transm., Distrib.*, vol. 4, pp. 1104–1115(11), Oct. 2010.
[31] G. Korres and N. Manousakis, "State estimation and observability analysis for phasor measurement unit measured systems," *IET Gener., Transm., Distrib.*, vol. 6, no. 9, pp. 902–913, Sep. 2012.
[32] M. Gol and A. Abur, "A hybrid state estimator for systems with limited number of PMUs," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1511–1517, May 2015.
[33] M. T. Heath, *Scientific Computing: An Introductory Survey*, E. M. Munson, Ed., 2nd ed. New York, NY, USA: McGraw-Hill Higher Education, 1996.
[34] J. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 1997.
[35] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
[36] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming With the Message-Passing Interface*, 3rd ed. Cambridge, MA, USA: MIT Press, Nov. 2014.
[37] B. Bode, M. Butler, T. Dunning, T. Hoefler, W. Kramer, W. Gropp, and W. Hwu, "The blue waters super-system for super-science," in *Contemporary High Performance Computing*, J. Vetter, Ed. London, U.K.: Chapman and Hall, 2013, pp. 339–366.
[38] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive and corrective actions," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4909–4917, Nov. 2013.
[39] J. P. Hespanha, Partition a MATLAB function for graph partitioning, Oct. 2004 [Online]. Available: http://www.ece.ucsb.edu/hespanha
[40] R. W. Cottle, J. Pang, and R. Stone, *Linear Complementarity Problem*. New York, NY, USA: Academic Press, 1992.
[41] A. Minot and N. Li, "A fully distributed state estimation using matrix splitting methods," in *Proc. Amer. Control Conf.*, 2015.
[42] R. S. Varga, *Matrix Iterative Analysis*, ser. Springer Series in Computational Mathematics. Berlin, Germany: Springer-Verlag, 2000.

Ariana Minot (S'11) received B.S. degrees in physics and in mathematics from Duke University in 2010 and M.Sc. degree in applied mathematics from Harvard University in 2014. She is currently a PhD candidate in applied mathematics at Harvard University, where she is a member of the Signals, Information, and Networks Group (SING).

Her research interests are in statistical inference and distributed optimization and estimation algorithms with applications for electric power systems.

Yue M. Lu (S'04–M'07–SM'12) received the M.Sc. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, both in 2007.

He was a Research Assistant at the University of Illinois at Urbana-Champaign, and was with Microsoft Research Asia, Beijing, China and Siemens Corporate Research, Princeton, NJ. From September 2007 to September 2010, he was a postdoctoral researcher at the Audiovisual Communications Laboratory at Ecole Polytechnique Fdrale de Lausanne (EPFL), Switzerland. He is currently an Assistant Professor of electrical engineering at Harvard University, Cambridge, MA, directing the Signals, Information, and Networks Group (SING) at the School of Engineering and Applied Sciences. His research interests are in the general areas of signal processing, statistical inference and imaging.

Prof. Lu received the Most Innovative Paper Award of IEEE International Conference on Image Processing (ICIP) in 2006 for his paper (with Minh N. Do) on the construction of directional multiresolution image representations, and the Best Student Paper Award of IEEE ICIP in 2007. He also coauthored a paper (with Ivan Dokmanic and Martin Vetterli) that won the Best Student Paper Award of IEEE International Conference on Acoustics, Speech and Signal Processing in 2011.

Na Li (M'09) received her B.S. degree in mathematics and applied mathematics from Zhejiang University in China and her PhD degree in Control and Dynamical systems from the California Institute of Technology in 2013.

She is an Assistant Professor in the School of Engineering and Applied Sciences in Harvard University. She was a postdoctoral associate of the Laboratory for Information and Decision Systems at Massachusetts Institute of Technology. She was a Best Student Paper Award finalist in the 2011 IEEE Conference on Decision and Control. Her research lies in the design, analysis, optimization and control of distributed network systems, with particular applications to power networks and systems biology/physiology.