



# Distributed weighted least-squares estimation with fast convergence for large-scale systems<sup>☆</sup>



Damián Edgardo Marelli<sup>a,c</sup>, Minyue Fu<sup>a,b</sup>

<sup>a</sup> School of Electrical Engineering and Computer Science, University of Newcastle, University Drive, Callaghan, NSW 2308, Australia

<sup>b</sup> Department of Control Science and Engineering and State Key Laboratory of Industrial Control Technology, Zhejiang University, 388 Yuhangtang Road Hangzhou, Zhejiang Province, 310058, PR China

<sup>c</sup> Acoustics Research Institute, Austrian Academy of Sciences, Austria

## ARTICLE INFO

### Article history:

Received 29 December 2013

Received in revised form

9 August 2014

Accepted 5 September 2014

Available online 17 November 2014

### Keywords:

Distributed estimation

Distributed state estimation

Large scale optimization

Sensor network

Networked control

## ABSTRACT

In this paper we study a distributed weighted least-squares estimation problem for a large-scale system consisting of a network of interconnected sub-systems. Each sub-system is concerned with a subset of the unknown parameters and has a measurement linear in the unknown parameters with additive noise. The distributed estimation task is for each sub-system to compute the globally optimal estimate of its own parameters using its own measurement and information shared with the network through neighborhood communication. We first provide a fully distributed iterative algorithm to asymptotically compute the global optimal estimate. The convergence rate of the algorithm will be maximized using a scaling parameter and a preconditioning method. This algorithm works for a general network. For a network without loops, we also provide a different iterative algorithm to compute the global optimal estimate which converges in a finite number of steps. We include numerical experiments to illustrate the performances of the proposed methods.

© 2014 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/3.0/>).

## 1. Introduction

A sensor network is a web of a large number of intelligent sensing and computing nodes connected via a communication network (Dargie & Poellabauer, 2010). The emergence of sensor networks calls for the development of distributed algorithms for a number of tasks to replace the traditional centralized methods. In particular, the development of distributed algorithms for parameter estimation has recently attracted a great deal of attention (Fang & Li, 2008; Kar, Moura, & Ramanan, 2012; Li & AlRegib, 2007, 2009; Lopes & Sayed, 2008; Ribeiro & Giannakis, 2006a,b; Ribeiro, Schizas, Rousmeliotis, & Giannakis, 2010; Xiao, Ribeiro, Luo, & Giannakis, 2006). They find applications in environmental and

weather monitoring, industrial process monitoring and control, surveillance, state estimation for smart grid, etc.

Existing distributed estimation methods can be classified in several ways. The first classification of the methods is done according to whether a coordinating node or fusion center (FC) is present. When an FC is present, each node communicates with the FC either directly (via a star topology) or indirectly (via a mesh topology), i.e., there is a directed communication path from either node to the FC. The estimation is carried out at the FC, after some possible pre-processing at each node before transmission. The methods proposed in Fang and Li (2008), Li and AlRegib (2007, 2009), Ribeiro and Giannakis (2006a,b) are of this type. On the other hand, when no FC is present, estimation is done by executing a cooperative algorithm among all the nodes of the network. The network is connected in some way and each node communicates with its neighboring nodes only. Representative methods of this type include Conejo, de la Torre, and Canas (2007), Gómez-Expósito, Villa Jaén, Gómez-Quiles, Rousseaux, and Van Cutsem (2011), Kar et al. (2012) and Lopes and Sayed (2008). A second classification is done by whether the estimation method is static or dynamic. In static estimation, a set of parameters is estimated using the measurements of all nodes, which collectively form a snapshot of the system at a fixed time. Examples of this type include Fang and Li

<sup>☆</sup> This work was partially supported by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1414) and by the Austrian Science Fund (FWF project M1230-N13). The material in this paper was partially presented at the 52nd IEEE Conference on Decision and Control (CDC 2013), December 10–13, 2013, Florence, Italy. This paper was recommended for publication in revised form by Associate Editor Tongwen Chen under the direction of Editor Ian R. Petersen.

E-mail addresses: [Damian.Marelli@newcastle.edu.au](mailto:Damian.Marelli@newcastle.edu.au) (D.E. Marelli), [Minyue.Fu@newcastle.edu.au](mailto:Minyue.Fu@newcastle.edu.au) (M. Fu).

<http://dx.doi.org/10.1016/j.automatica.2014.10.077>

0005-1098/© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).

(2008), Kar et al. (2012), Li and AlRegib (2007, 2009), Lopes and Sayed (2008) and Ribeiro and Giannakis (2006a,b). On the other hand, in dynamic estimation all the nodes track the evolution of a set of parameters for which a dynamic model is available, as in the Refs. Carli, Chiuso, Schenato, and Zampieri (2008), Hlinka, Sluciak, Hlawatsch, Djuric, and Rupp (2012), Khan and Moura (2008) and Ribeiro et al. (2010). Some “hybrid” methods exist, which permit tracking a time-varying sequence of parameters, without a dynamic model, by updating a static estimate at each time (Lopes & Sayed, 2008). A final classification is done by whether a distributed estimation method is a small-scale one or a large-scale one. In a small-scale method, all nodes estimate a common set of parameters (Fang & Li, 2008; Kar et al., 2012; Li & AlRegib, 2007, 2009; Lopes & Sayed, 2008; Ribeiro & Giannakis, 2006a,b). But in a large-scale method, each node only reconstructs a subset of the parameters, i.e., the collective knowledge of the parameters is distributed among all the nodes (Conejo et al., 2007; Gómez-Expósito et al., 2011; Huang, Werner, Huang, Kashyap, & Gupta, 2012; Khan & Moura, 2008). Large-scale estimation is in general more challenging than its small-scale counterpart.

In this paper we study distributed static estimation for a large-scale system consisting of a network of interconnected sub-systems. Each sub-system is concerned with a subset of the unknown parameters and has measurements, linear in the unknown parameters, corrupted by additive noise. The distributed estimation task is for each sub-system to estimate its local state using its own measurement and information shared with the network through neighborhood communication. We use the weighted least squares (WLS) criterion for optimal estimation. The goal is that the composite estimate of the whole system, consisting of all local estimates, will become globally optimal in the sense that it is the same as the optimal estimate obtained using all the measurements and a centralized estimation method.

This problem is motivated by many applications involving a large-scale spatially distributed system. For example, the state estimation problem for a large power network is concerned with estimating the voltages and phases of the power supply at each sub-system, consisting of a number of buses or a substation, using measurements (provided by, for example, a phasor measurement unit (PMU) or a supervisory control and data acquisition (SCADA) system) (Conejo et al., 2007; Jiang, Vittal, & Heydt, 2008). Interactions of sub-systems are reflected by the fact that local measurements available at each sub-system typically involve neighboring sub-systems. For example, a current measurement at a conjunction depends on the voltage difference of two neighboring buses. In a smart grid setting, each sub-system is only interested in the local state, i.e., its own voltages and phases, using local measurements and information acquired from neighboring sub-systems via neighborhood communication (Tai, Marelli, Rohr, & Fu, 2013). For a large power network, it is both impractical and unnecessary for every sub-system to estimate the whole state of the system, thus distributed estimation methods for local state estimation are naturally called for. The second example is the localization problem for a wireless sensor network, involving estimating the locations of all sensors using relative measurements (e.g., relative distances or relative positions) between sensors (Diao, Fu, & Zhang, 2013; Khan, Kar, & Moura, 2009). For a small sensor network with a few sensing nodes, it is possible to aggregate all the measurements at an FC to compute a global estimate of all sensor locations. Such an algorithm is not scalable, and would require massive computing resources when the network becomes large. It is also unnecessary for each sensor to localize other nodes. A distributed method is preferred, where each node aims to estimate its own location using local measurements and neighborhood communication. The third example is a traffic network for a city or a highway system, where each node or sub-system wants to estimate its local state

(e.g., traffic flow rates, delays, etc.). Due to the spatial correlations of the traffic flows in different sub-systems, neighboring traffic information is certainly useful in predicting the traffic conditions at each sub-system. Again, distributed estimation methods are preferred over centralized methods. Many other examples in sensor networks can be found in, e.g., Ribeiro et al. (2010), Kar et al. (2012) and Yang et al. (2010).

We first provide a fully distributed iterative algorithm for each node to compute its own local estimate. This algorithm works for a general connected network. Contrary to the method proposed in Conejo et al. (2007), our algorithm guarantees that the composite estimate will converge asymptotically to the global WLS estimate. We then focus on the convergence rate of the algorithm. Since our method is based on Richardson’s method for solving systems of linear equations (Bertsekas & Tsitsiklis, 1997), its convergence rate depends on certain scaling parameter and a preconditioning matrix. Choosing the optimum scaling parameter requires the knowledge of the largest and the smallest eigenvalues of certain positive definite matrix (the estimation error covariance). A distributed algorithm for estimating these values can be obtained using the power method (Bertsekas & Tsitsiklis, 1997). However, to prevent numerical instability, this approach requires periodically executing a normalization step, which needs to be carried out in a distributed manner. In Yang et al. (2010), this is done using average consensus. A drawback of this approach is that consensus itself converges asymptotically. This significantly slows down the convergence of the eigenvalue estimation. To avoid this, we propose a different method in which normalization is done locally, at each node, without inter-node communication. In this way, the optimal scaling parameter can be obtained using a distributed method. We then address the problem of designing the preconditioning matrix. Our distributed scenario constrains us to use a block diagonal matrix. Choosing the optimal matrix under this constraint, and using only distributed processing, is a very challenging problem for which we are not able to provide a solution. Nevertheless, we are able to bound the difference between the convergence rate achieved using the optimal matrix, and the convergence rate resulting using a practically feasible matrix design. This bound turns out to have a simple expression which depends on the network connectivity. A shortened version of this method appears in the conference paper (Marelli & Fu, 2013).

For an acyclic network (i.e., its communication graph contains no loops), we provide a different iterative algorithm for distributed estimation. As opposed to the previous algorithm, in this one, the composite estimate is guaranteed to converge to the globally optimal estimate in a finite number of steps. Indeed, we show that the convergence time equals the diameter of the aforementioned graph. Numerical experiments show that this method offers a major reduction in convergence time.

The rest of paper is organized as follows. In Section 2 we describe the distributed WLS estimation the problem. In Section 3 we derive the first distributed WLS method, which converges asymptotically. In Section 3.1 we describe distributed methods for finding the value of the scaling parameter which yields the fastest convergence rate. In Section 3.2 we describe a sub-optimal choice for preconditioning matrix, and we bound its sub-optimality. In Section 4 we introduce the second distributed WLS method which converges in finite time. Numerical experiments are presented in Section 5, and concluding remarks are given in Section 6. For the ease of readability, all proofs are contained in the Appendix.

**Notation 1.** For a vector  $x$ ,  $\|x\|$  denotes its 2-norm, and  $[x]_i$  denotes its  $i$ th entry. For a matrix  $X$ ,  $\|X\|$  denotes its operator (induced) norm. For square symmetric matrices  $X$  and  $Y$ ,  $X < Y$  means that matrix  $Y - X$  is positive-definite. For vectors and matrices, the superscript  $T$  denotes its transpose, and  $*$  denotes its transpose conjugate.

## 2. Problem description

Consider a network formed by  $I$  nodes. For each  $i = 1, \dots, I$ , Node  $i$  has an associated parameter vector  $x_i \in \mathbb{C}^{d_i}$ , and measures the vector  $y_i \in \mathbb{C}^{p_i}$ , which is given by

$$y_i = \sum_{j=1}^I A_{i,j} x_j + v_i, \quad (1)$$

where  $v_i \sim \mathcal{N}(0, R_i)$  denotes the measurement noise. The noises  $v_i$  and  $v_j$  are statistically independent, whenever  $i \neq j$ .

Let  $x^T = [x_1^T, \dots, x_I^T]$ ,  $y^T = [y_1^T, \dots, y_I^T]$ ,  $v^T = [v_1^T, \dots, v_I^T]$ ,  $R = \text{diag}\{R_1, \dots, R_I\}$  and  $A = [A_{i,j}]_{i,j=1,\dots,I}$ . Then, we can write the measurement model of the whole network as

$$y = Ax + v, \quad (2)$$

with  $v \sim \mathcal{N}(0, R)$ . The WLS estimator  $\hat{x}$  of  $x$  is defined by Kay (1993, Eq. (8.14))

$$\hat{x} = \arg \min_x (y - Ax)^* R^{-1} (y - Ax).$$

Its solution is given by

$$\hat{x} = \Psi^{-1} \alpha \quad (3)$$

with

$$\alpha = A^* R^{-1} y \quad \text{and} \quad \Psi = A^* R^{-1} A.$$

For the WLS problem to be well defined, we make the following assumption:

**Assumption 2.** Matrix  $A$  has full column rank and  $R$  is non-singular.

Computing (3) requires global information, i.e., all the measurements and the information about  $A$  and  $R$  need to be made available together. Our goal is to derive distributed methods in which Node  $i$  computes the component  $\hat{x}_i$  of the estimate  $\hat{x}$ , corresponding to  $x_i$ , using only the local measurement  $y_i$  and information received from its neighbors (a formal definition of neighborhood will be given later).

In the rest of the paper we use the following notation:

**Notation 3.** Let  $\mathcal{I}_i = \{j : A_{j,i} \neq 0\}$  denote the set of nodes whose measurements involve the parameters of Node  $i$ , and  $\mathcal{O}_i = \{j : A_{i,j} \neq 0\}$  denote the set of nodes whose parameters are involved in the measurements of Node  $i$ . Let  $\mathcal{N}_i = \mathcal{I}_i \cup \mathcal{O}_i$  be the neighborhood of Node  $i$ . We also define  $\mathcal{B}_i = \{j : \mathcal{N}_i \cap \mathcal{N}_j \neq \emptyset\}$  to be the set of nodes whose neighborhood have a non-empty intersection with that of Node  $i$ . Notice that  $\mathcal{B}_i = \{j : \Psi_{i,j} \neq 0\}$ .

## 3. Asymptotic method for WLS estimation

The distributed WLS method derived in this section uses the following definition of neighbor node.

**Definition 4.** Node  $j$  is a neighbor of Node  $i$  if  $j \in \mathcal{N}_i$ .

Also, the proposed method requires the following connectivity assumption:

**Assumption 5.** For each  $i = 1, \dots, I$ , Node  $i$  can send/receive information to/from all its neighbors. Also,  $A_{i,j}$ , for all  $j \in \mathcal{O}_i$ , and  $R_i$  are available at Node  $i$ .

Although our method works regardless of whether the network is sparse or not, it works most efficiently for sparse networks. A network is called sparse if the cardinality of  $\mathcal{N}_i$  is small for all  $i = 1, 2, \dots, I$ .

Consider any block diagonal positive definite matrix

$$\Pi = \text{diag}\{\Pi_1, \dots, \Pi_I\} \quad (4)$$

with  $\Pi_i \in \mathbb{C}^{d_i \times d_i}$ ,  $i = 1, \dots, I$ . Then define

$$\Upsilon = \Pi^{1/2} \Psi \Pi^{1/2} \quad (5)$$

and choose

$$0 < \gamma < 2 \|\Upsilon\|^{-1}. \quad (6)$$

Let  $\tilde{\alpha} = (\gamma \Pi)^{1/2} \alpha$  and  $\tilde{\hat{x}} = (\gamma \Pi)^{-1/2} \hat{x}$ . From (3) we have

$$\tilde{\hat{x}} = (\gamma \Upsilon)^{-1} \tilde{\alpha}.$$

From (6),  $0 < \gamma \Upsilon < 2I$ . Hence,  $-I < \gamma \Upsilon - I < I$  and therefore  $\|I - \gamma \Upsilon\| < 1$ . In view of this, we can use Richardson's method (Bertsekas & Tsitsiklis, 1997) to compute  $\tilde{\hat{x}}$  recursively. This leads to

$$\tilde{\hat{x}}(t+1) = (I - \gamma \Upsilon) \tilde{\hat{x}}(t) + \tilde{\alpha}. \quad (7)$$

Then, by substituting the expressions of  $\tilde{\alpha}$  and  $\tilde{\hat{x}}$ , we obtain straightforwardly

$$\hat{x}(t+1) = (I - \gamma \Pi \Psi) \hat{x}(t) + \gamma \Pi \alpha. \quad (8)$$

We call  $\Pi$  the preconditioning matrix, because, as it will be explained in Section 3.2, it is used to increase the convergence rate of the recursions (8).

Let

$$\alpha_i = \sum_{k \in \mathcal{I}_i} \alpha_i^{(k)}, \quad (9)$$

with  $\alpha_i^{(k)} = A_{k,i}^* R_k^{-1} y_k$ , for  $k = 1, \dots, I$ , so that  $\alpha^T = [\alpha_1^T, \dots, \alpha_I^T]$ . Also, for  $i, j = 1, \dots, I$ , let

$$\Psi_{i,j} = \sum_{k: i,j \in \mathcal{O}_k} \Psi_{i,j}^{(k)}, \quad (10)$$

where  $\Psi_{i,j}^{(k)} = A_{k,i}^* R_k^{-1} A_{k,j}$ , for all  $k = 1, \dots, I$ , so that  $\Psi = [\Psi_{i,j}]_{i,j=1,\dots,I}$ . We have that

$$\begin{aligned} [\Psi \hat{x}(t)]_i &= \sum_{j=1}^I \Psi_{i,j} \hat{x}_j(t) \\ &= \sum_{j=1}^I \sum_{k: i,j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t) \\ &= \sum_{k: i \in \mathcal{O}_k} \sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t) \\ &= \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t). \end{aligned} \quad (11)$$

Then, from (8), (11) and (9), we obtain

$$\begin{aligned} \hat{x}_i(t+1) &= \hat{x}_i(t) - \gamma \Pi_i \sum_{j=1}^I \Psi_{i,j} \hat{x}_j(t) + \gamma \Pi_i \alpha_i \\ &= \hat{x}_i(t) - \gamma \Pi_i \left( \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t) - \sum_{k \in \mathcal{I}_i} \alpha_i^{(k)} \right). \end{aligned} \quad (12)$$

Notice that the matrices  $\Psi_{i,j}^{(k)}$  are only available at Node  $k$ . That is, Node  $k$  acts as an intermediary between Node  $j$ , which transmits

$\hat{x}_j(t)$ , and Node  $i$ , which receives  $\sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t)$ . This means that node  $j$  should transmit  $\hat{x}_j(t)$  to all nodes  $k$  with  $j \in \mathcal{O}_k$ , or equivalently, to all nodes in  $\mathcal{J}_j$ . However, Node  $j$  does not know which nodes are in  $\mathcal{J}_j$ . Thus, Node  $j$  simply transmits  $\hat{x}_j(t)$  to all nodes in  $\mathcal{N}_j$ , and it is up to the receiving Node  $k$  to detect whether Node  $j \in \mathcal{O}_k$  or not.

Following the discussion above, we obtain the following algorithm to implement (12).

---

**Algorithm 1 - distributed WLS estimation:**

**Initialization:**

- (1) For each  $k = 1, \dots, I$  and  $i \in \mathcal{O}_k$ , Node  $k$  computes  $\alpha_i^{(k)}$  and sends it to Node  $i$ .
- (2) On reception, Node  $i$  computes  $\alpha_i = \sum_{k \in \mathcal{I}_i} \alpha_i^{(k)}$ .
- (3) For each  $i = 1, \dots, I$ , Node  $i$  sets  $\hat{x}_i(1) = 0$ .

**Main loop:** At time  $t \in \mathbb{N}$ :

- (1) For each  $j = 1, \dots, I$  and  $k \in \mathcal{N}_j$ , Node  $j$  sends its current estimate  $\hat{x}_j(t)$  to Node  $k$ .
- (2) On reception, for each  $k = 1, \dots, I$  and  $i \in \mathcal{O}_k$ , Node  $k$  sends to Node  $i$

$$\check{x}_{i,k}(t) = \sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \hat{x}_j(t).$$

- (3) On reception, for each  $i = 1, \dots, I$ , Node  $i$  computes

$$\hat{x}_i(t+1) = \hat{x}_i(t) - \gamma \Pi_i \left( \sum_{k \in \mathcal{I}_i} \check{x}_{i,k}(t) - \alpha_i \right).$$

---

To implement Algorithm 1, we need to design the scaling factor  $\gamma$  and the preconditioning matrices  $\Pi_i$ , for all  $i = 1, \dots, I$ . We address these two tasks in Sections 3.1 and 3.2, respectively.

### 3.1. Distributed design of the scaling factor $\gamma$

In this section we study two approaches for designing the scaling factor  $\gamma$ . In Section 3.1.1 we describe a distributed algorithm which converges asymptotically to the optimal value of  $\gamma$ , i.e., the resulting  $\gamma$  will achieve the maximum convergence speed. In Section 3.1.2, we give another distributed algorithm which converges in finite time to a sub-optimal value of  $\gamma$ .

#### 3.1.1. Asymptotic algorithm for $\gamma$

In view of (7), the value of  $\gamma$  that maximizes the convergence rate is

$$\gamma = \frac{2}{\|\mathcal{Y}\| + \|\mathcal{Y}^{-1}\|^{-1}}, \quad (13)$$

because this is the value that minimizes  $\|I - \gamma\mathcal{Y}\|$ . In order for each node to find the value of  $\gamma$  given in (13), we need distributed methods for finding  $\|\mathcal{Y}\|$  and  $\|\mathcal{Y}^{-1}\|^{-1}$ . We give these methods below. These methods yield, at Node  $i$  and time step  $t$ , estimates  $\overline{\gamma}_i(t)$  and  $\underline{\gamma}_i(t)$ , of  $\|\mathcal{Y}\|$  and  $\|\mathcal{Y}^{-1}\|^{-1}$ , respectively. Then, at the same node and time step, the estimate  $\gamma_i(t)$  of  $\gamma$  is obtained by

$$\gamma_i(t) = \frac{2}{\overline{\gamma}_i(t) + \underline{\gamma}_i(t)}.$$

**Distributed method for finding  $\|\mathcal{Y}\|$ :**

Choose any vector  $b(0) \neq 0$  and let  $b(t) = \mathcal{Y}^t b(0)$ . Then, using (11) with  $\Pi^{1/2} b(t)$  in place of  $\hat{x}(t)$ , we obtain at Node  $i$ ,

$$\begin{aligned} b_i(t+1) &= [\mathcal{Y}b(t)]_i \\ &= \Pi_i^{1/2} [\Psi \Pi^{1/2} b(t)]_i \\ &= \Pi_i^{1/2} \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} \Psi_{i,j}^{(k)} \Pi_j^{1/2} b_j(t), \end{aligned} \quad (14)$$

where  $b_i(t)$  denotes the  $i$ th block component of  $b(t)$ . Then, using the power method (Bertsekas & Tsitsiklis, 1997), Node  $i$  can asymptotically compute  $\|\mathcal{Y}\|$  as follows

$$\|\mathcal{Y}\| = \lim_{t \rightarrow \infty} \frac{\|b_i(t)\|}{\|b_i(t-1)\|}. \quad (15)$$

An inconvenience with the approach above is that  $b(t)$  either increases or decreases indefinitely. To avoid this, the vector  $b(t)$  can be periodically normalized. In Yang et al. (2010), this was done using average consensus (in the continuous-time case). As we mentioned in Section 1, we avoid the drawbacks of that method by providing an alternative approach in which  $b(t)$  is normalized at each node, and each iteration, without inter-node communication. This algorithm is given below:

---

**Algorithm 2 - distributed estimation of  $\|\mathcal{Y}\|$ :** For each  $k = 1, \dots, I$ , Node  $k$ , chooses  $\bar{b}_k(1)$ , with  $\|\bar{b}_k(1)\| = 1$  and sets  $\varsigma_k(1) = 1$  and  $v_{i,j}^{(k)}(1) = 1$ , for all  $i, j \in \mathcal{N}_k$ . Then, at time  $t \in \mathbb{N}$ :

- (1) For each  $j = 1, \dots, I$  and  $k \in \mathcal{N}_j$ , Node  $j$  sends  $(\Pi_j^{1/2} \bar{b}_j(t), \varsigma_j(t))$  to Node  $k$ .
- (2) On reception, for each  $k = 1, \dots, I$  and  $i \in \mathcal{O}_k$ , Node  $k$  sends  $(\check{b}_i^{(k)}(t), \bar{\varsigma}_i^{(k)}(t))$  to Node  $i$ , where

$$\check{b}_i^{(k)}(t) = \sum_{j \in \mathcal{O}_k} v_{i,j}^{(k)}(t) \Psi_{i,j}^{(k)} \Pi_j^{1/2} \bar{b}_j(t),$$

$$\bar{\varsigma}_i^{(k)}(t) = \max_{j \in \mathcal{N}_k} v_{i,j}^{(k)}(t),$$

and

$$v_{i,j}^{(k)}(t) = \frac{\varsigma_i(t)}{\varsigma_j(t)} v_{i,j}^{(k)}(t-1).$$

- (3) On reception, for each  $i = 1, \dots, I$ , Node  $i$  computes

$$\bar{b}_i(t+1) = \varsigma_i(t+1) \check{b}_i(t+1),$$

$$\varsigma_i(t+1) = \max \left\{ \|\check{b}_i(t+1)\|, \bar{\varsigma}_i^{(k)}(t), k \in \mathcal{I}_i \right\}^{-1},$$

with

$$\check{b}_i(t+1) = \Pi_i^{1/2} \sum_{k \in \mathcal{I}_i} \check{b}_i^{(k)}(t). \quad (16)$$

Also, the estimate  $\overline{\gamma}_i(t)$  of  $\|\mathcal{Y}\|$  is

$$\overline{\gamma}_i(t) = \varsigma_i(t+1)^{-1}. \quad (17)$$

---

The convergence of Algorithm 2 to  $\|\mathcal{Y}\|$  is guaranteed by the next theorem.

**Theorem 6.** Consider the network (1) together with Assumptions 2 and 5. Then, for each  $i \in \{1, \dots, I\}$ ,

$$\lim_{t \rightarrow \infty} \overline{\gamma}_i(t) = \|\mathcal{Y}\|. \quad (18)$$

**Distributed method for finding  $\|\mathcal{Y}^{-1}\|^{-1}$ :**

Let  $c \geq \|\mathcal{Y}\|$  and  $\Phi = cl - \mathcal{Y}$ . It follows that

$$\begin{aligned} \|\mathcal{Y}^{-1}\|^{-1} &= \underline{\text{eig}}(\mathcal{Y}) \\ &= c - \overline{\text{eig}}(\Phi) \\ &= c - \|\Phi\|, \end{aligned}$$

where, for a symmetric matrix  $X$ ,  $\underline{\text{eig}}(X)$  and  $\overline{\text{eig}}(X)$  denote the smallest and largest eigenvalues of  $X$ , respectively. Hence, we can

find  $\|\mathcal{Y}^{-1}\|^{-1}$  by applying Algorithm 2 on  $\Phi$ , to find  $\|\Phi\|$ . To this end, at Node  $i$  and time  $t$ , we choose  $c = \bar{\mathcal{Y}}_i(t)$ . This leads to the following algorithm:

**Algorithm 3 - distributed estimation** of  $\|\mathcal{Y}^{-1}\|^{-1}$ : Apply Algorithm 2 with (16) replaced by

$$\tilde{b}_i(t+1) = \bar{\mathcal{Y}}_i(t)\tilde{b}_i(t) - \Pi_i^{1/2} \sum_{k \in \mathcal{I}_i} \check{b}_{i,k}(t),$$

and (17) replaced by

$$\begin{aligned} \underline{\mathcal{Y}}_i(t) &= \bar{\mathcal{Y}}_i(t) - \bar{\Phi}_i(t), \\ \bar{\Phi}_i(t) &= \zeta_i(t+1)^{-1}. \end{aligned}$$

### 3.1.2. Finite-time algorithm for $\gamma$

A sub-optimal design of  $\gamma$  can be achieved using the following result.

**Theorem 7.** Condition (6) is satisfied by choosing  $\gamma$  so that

$$0 < \gamma < \frac{2}{\max_i \phi_i},$$

where

$$\begin{aligned} \phi_i &= \sum_{k \in \mathcal{I}_i} \nu_{i,k}, \\ \nu_{i,k} &= \sum_{j \in \mathcal{O}_k} \left\| \Pi_i^{1/2} \Psi_{i,j}^{(k)} \Pi_j^{1/2} \right\|. \end{aligned}$$

The design of  $\gamma$  using Theorem 7 requires the global information  $\max_i \phi_i$ . For each,  $i = 1, \dots, I$ , Node  $i$  can obtain  $\phi_i$  from an initialization stage in which it receives  $\nu_{i,k}$ , from each Node  $k$ , with  $k \in \mathcal{I}_i$ . Then,  $\max_i \phi_i$  can be obtained by running the max-consensus algorithm (Olfati-Saber & Murray, 2004), in parallel with the estimation Algorithm 1. Notice that the max-consensus algorithm converges in finite time.

### 3.2. Design of the preconditioning matrix $\Pi$

As mentioned above, for a given choice of  $\mathcal{Y}$ , the fastest convergence rate of Algorithm 1 is achieved when  $\gamma$  is chosen as in (13). Under this choice of  $\gamma$ , we have that

$$\begin{aligned} \|I - \gamma \mathcal{Y}\| &= \gamma \|\mathcal{Y}\| - 1 \\ &= \frac{2 \|\mathcal{Y}\|}{\|\mathcal{Y}\| + \|\mathcal{Y}^{-1}\|^{-1}} - 1 \\ &= \frac{\|\mathcal{Y}\| - \|\mathcal{Y}^{-1}\|^{-1}}{\|\mathcal{Y}\| + \|\mathcal{Y}^{-1}\|^{-1}} \\ &= \frac{\kappa(\mathcal{Y}) - 1}{\kappa(\mathcal{Y}) + 1}, \end{aligned}$$

where  $\kappa(\mathcal{Y}) = \|\mathcal{Y}\| \|\mathcal{Y}^{-1}\|$  denotes the condition number of  $\mathcal{Y}$ . Then, from (7), there exists  $K \geq 0$ , such that

$$\begin{aligned} \|\hat{x} - \hat{x}(t)\| &\leq K \|I - \gamma \mathcal{Y}\|^t \\ &= K \exp \left\{ t \log \frac{\kappa(\mathcal{Y}) - 1}{\kappa(\mathcal{Y}) + 1} \right\}, \end{aligned}$$

where we recall that  $\hat{x}$  denotes the global estimate of  $x$ , given by (3). Then, we define the time constant  $\tau(\mathcal{Y})$  of the distributed WLS algorithm by

$$\tau(\mathcal{Y}) = \frac{1}{\log \frac{\kappa(\mathcal{Y})+1}{\kappa(\mathcal{Y})-1}}. \quad (19)$$

Hence, a natural question is whether the preconditioning matrices  $\Pi_i$ ,  $i = 1, \dots, I$ , can be chosen so that  $\tau(\mathcal{Y})$  is minimized. While we are not able to answer this question, we have the following result, which follows using an argument similar to the one in Demmel (1983, Theorem 2).

**Theorem 8.** If  $\Pi_i = \Psi_{i,i}^{-1}$ , for all  $i = 1, \dots, I$ , then

$$\kappa(\mathcal{Y}) \leq \beta \kappa_*,$$

where

$$\begin{aligned} \beta &= \max_i |\mathcal{B}_i|, \\ \kappa_* &= \min_{\tilde{\Pi} \in \mathcal{P}} \kappa(\tilde{\Pi}^{1/2} \Psi \tilde{\Pi}^{1/2}), \end{aligned}$$

with  $\mathcal{P}$  denoting the set of positive definite block diagonal matrices of the form (4).

Theorem 8 states that, if the preconditioning matrices  $\Pi_i$ ,  $i = 1, \dots, I$ , are chosen as

$$\Pi_i = \Psi_{i,i}^{-1} \quad (20)$$

then  $\kappa(\mathcal{Y})$  is at most  $\beta$  times bigger than the smallest possible value  $\kappa_*$  achievable using block diagonal preconditioning matrices. Notice that  $\mathcal{B}_i = \{j : \mathcal{I}_i \cap \mathcal{I}_j \neq \emptyset\} \subseteq \{j : \mathcal{N}_i \cap \mathcal{N}_j \neq \emptyset\}$ . Hence,  $\beta$  is bounded by the maximum number of two-hop neighbors over the whole network. Hence, it does not necessarily grow with the network size.

Now, we have

$$\lim_{\kappa \rightarrow \infty} \kappa \log \left( \frac{\kappa + 1}{\kappa - 1} \right) = 2.$$

Hence, from Theorem 8, for large  $\kappa(\mathcal{Y})$ , we have

$$\begin{aligned} \tau(\mathcal{Y}) &\simeq \frac{\kappa(\mathcal{Y})}{2} \\ &\leq \frac{\beta}{2} \min_{\tilde{\Pi}} \kappa(\tilde{\Pi}^{1/2} \Psi \tilde{\Pi}^{1/2}) \\ &\simeq \beta \tau_*, \end{aligned} \quad (21)$$

where

$$\tau_* = \min_{\tilde{\Pi} \in \mathcal{P}} \tau(\tilde{\Pi}^{1/2} \Psi \tilde{\Pi}^{1/2}).$$

Hence, if  $\Pi_i$ ,  $i = 1, \dots, I$ , are chosen as in (20), and  $\kappa(\mathcal{Y})$  is large, then the time constant  $\tau(\mathcal{Y})$  is at most  $\beta$  times bigger than the minimum value  $\tau_*$ .

**Remark 9.** In view of (20) and (10),

$$\Pi_i = \left( \sum_{k \in \mathcal{I}_i} \Psi_{i,i}^{(k)} \right)^{-1}.$$

Hence, its computation requires the matrices  $\Psi_{i,i}^{(k)}$ ,  $k \in \mathcal{I}_i$ , to be transmitted from Node  $k$  to Node  $i$  during an initialization stage.

## 4. Finite-time method for WLS estimation

In this method we replace the definition of neighborhood by the following one:

**Definition 10.** Node  $j$  is a neighbor of Node  $i$  if  $j \in \mathcal{B}_i$ .

Consequently, we replace the connectivity Assumption 5 by the following one:

**Assumption 11.** For each  $i = 1, \dots, I$ , Node  $i$  can send/receive information to/from all its neighbors. Also,  $\Psi_{j,i}$ , for all  $j \in \mathcal{B}_i$ , and  $\alpha_i$  are available at Node  $i$ .

To illustrate the idea behind the proposed algorithm, we consider a network with two nodes. The next lemma states how to obtain the global optimal solution, at each node, in this simple case.

**Lemma 12.** Consider the network (1) together with Assumption 2. If there are only two nodes, labeled by  $a$  and  $b$ , then  $\Psi_{a,a} - \Psi_{a,b} \check{\Sigma}_b \Psi_{b,a}$  is an invertible matrix and the global estimate  $\hat{x}_a$  of the components  $x_a$  associated to Node  $a$  is given by

$$\hat{x}_a = \Sigma_a (\alpha_a - \Psi_{a,b} \check{x}_b),$$

$$\Sigma_a = \left( \Psi_{a,a} - \Psi_{a,b} \check{\Sigma}_b \Psi_{b,a} \right)^{-1},$$

where

$$\check{x}_b = \check{\Sigma}_b \alpha_b,$$

$$\check{\Sigma}_b = \Psi_{b,b}^{-1}.$$

Our next result is an immediate generalization of the one above, to a network with a star topology, i.e., in which all nodes are only possibly connected to a single one.

**Lemma 13.** Consider the network (1) together with Assumption 2. Suppose that  $\Psi_{j,k} = 0$ , for all  $j, k \in \{1, \dots, I\} \setminus \{i\}$  and  $j \neq k$  (i.e., all nodes are only possibly connected to Node  $i$ ). Then  $\Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Psi_{i,j} \check{\Sigma}_j \Psi_{j,i}$  is an invertible matrix and  $\hat{x}_i$  is given by

$$\hat{x}_i = \Sigma_i \left( \alpha_i - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Psi_{i,j} \check{x}_j \right),$$

$$\Sigma_i = \left( \Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Psi_{i,j} \check{\Sigma}_j \Psi_{j,i} \right)^{-1},$$

where

$$\check{x}_j = \check{\Sigma}_j \alpha_j,$$

$$\check{\Sigma}_j = \Psi_{j,j}^{-1}.$$

Then, using (11), (9) and Lemma 13, we obtain the following algorithm:

**Algorithm 4 - distributed WLS estimation:**

**Initialization:** For each  $i = 1, \dots, I$ ,

(1) Node  $i$  computes

$$\check{x}_i(0) = \check{\Sigma}_i(0) \alpha_i,$$

$$\check{\Sigma}_i(0) = \Psi_{i,i}^{-1}.$$

and for each  $j \in \mathcal{B}_i \setminus \{i\}$ ,

$$\check{x}_{i,j}(0) = \check{x}_i(0),$$

$$\check{\Sigma}_{i,j}(0) = \check{\Sigma}_i(0).$$

**Main loop:** For each  $i = 1, \dots, I$ , and time  $t \in \mathbb{N}$

(1) Node  $i$  computes, for each  $j \in \mathcal{B}_i \setminus \{i\}$ ,

$$\gamma_{i,j}(t) = \Psi_{j,i} \check{x}_i(t-1),$$

$$\Gamma_{i,j}(t) = \Psi_{j,i} \check{\Sigma}_i(t-1) \Psi_{i,j},$$

and sends  $(\gamma_{i,j}(t), \Gamma_{i,j}(t))$  to Node  $j$ .

(2) Node  $i$  computes

$$\check{x}_i(t) = \check{\Sigma}_i(t) \left( \alpha_i - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \gamma_{j,i}(t-1) \right),$$

$$\check{\Sigma}_i(t) = \left( \Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Gamma_{j,i}(t-1) \right)^{-1},$$

and, for each  $j \in \mathcal{B}_i \setminus \{i\}$ ,

$$\check{x}_{i,j}(t) = \check{\Sigma}_{i,j}(t) \left( \alpha_i - \sum_{j \in \mathcal{B}_i \setminus \{i,j\}} \gamma_{j,i}(t-1) \right),$$

$$\check{\Sigma}_{i,j}(t) = \left( \Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i,j\}} \Gamma_{j,i}(t-1) \right)^{-1}.$$

Our next step is to show that Algorithm 4 converges in finite time to the global WLS solution.

**Definition 14.** Each pair  $(i, j)$ ,  $i, j \in \{1, \dots, I\}$ , is called an *edge* if  $\Psi_{i,j} \neq 0$ . A *path* is a concatenation of contiguous edges, and its *length* is the number of edges forming it. For each  $i, j \in \{1, \dots, I\}$ , the *distance*  $d_{i,j}$  between Nodes  $i$  and  $j$  is defined as the minimum length of a path joining these two nodes. The *radius*  $\rho_i$  of Node  $i$  is defined as the maximum distance between Node  $i$  and any other node in the network. The *diameter* of the network is the maximum radius between all its nodes. A network is called *acyclic* if it does not contain a path forming a cycle.

The next theorem states that, if the network is acyclic, then the algorithm above yields the global estimate at each node in finite time.

**Theorem 15.** Consider the network (1) together with Assumptions 2 and 11. If the network is acyclic, then, for each  $i \in \{1, \dots, I\}$ ,  $j \in \mathcal{B}_i \setminus \{i\}$  and  $t \in \mathbb{N}$ , the matrices  $\Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Gamma_{j,i}(t-1)$  and  $\Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i,j\}} \Gamma_{j,i}(t-1)$  are invertible, and for all  $t \geq \rho_i$ ,

$$\check{x}_i(t) = \hat{x}_i. \quad (22)$$

## 5. Simulations

### 5.1. State estimation in power systems

In the first simulation we use the proposed distributed methods for state estimation in smart electricity networks, involving multi-area interconnected power systems (Huang et al., 2012). To this end, we use the IEEE 118-bus test system, whose specifications are given in Christie (1993). The system's diagram is shown in Fig. 1, where buses are represented by circles and lines by edges. Some buses have a phasor measurement unit (PMU) installed. These buses are shown in gray. Each PMU measures the voltage of the bus where it is installed, as well as the currents of the lines attached to that bus. The goal is to estimate the state vector  $x$ , containing voltage (a complex phasor) at each bus. For the purposes of state estimation, the buses are clustered in nodes. Two clustering examples as shown in Tables 1 and 3.

Let  $P$  denote the number of PMUs in the whole system. For each  $p = 1, \dots, P$ , let  $L_p$  denote the number of lines attached to the bus where PMU  $p$  is installed. Let also  $B_p^T = [e_p^T, y_{p,1}^T, \dots, y_{p,L_p}^T]$ , where the vectors  $e_p$  and  $y_{p,l}$ ,  $l = 1, \dots, L_p$ , are defined such that  $e_p x$  is the voltage of the installation bus, and  $y_{p,l} x$  is the current of the  $p$ -th attached line (the value of  $y_{p,l}$  is taken from Christie,

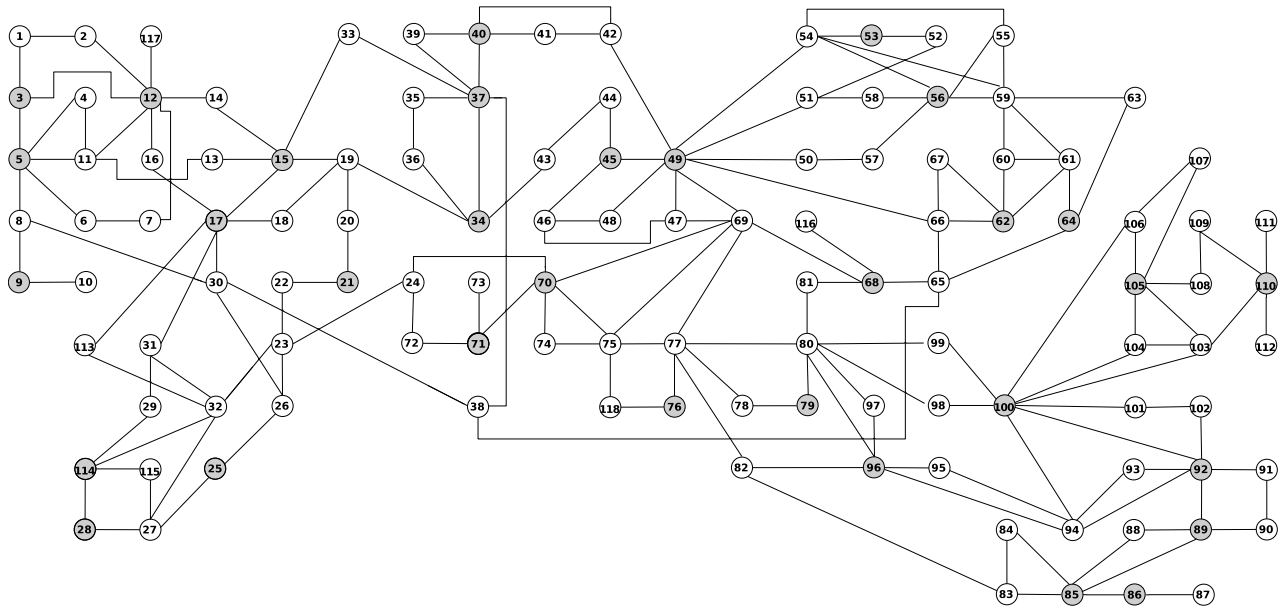


Fig. 1. Diagram of the IEEE 118-bus test system.

Table 1  
Nodes forming a cyclic network topology.

Node	Buses
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 117
2	13, 15, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 113, 114, 115
3	24, 38, 70, 71, 72, 73, 74
4	34, 35, 36, 37, 39, 40, 41, 42, 43
5	44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 77, 80, 81, 100, 116
6	75, 76, 78, 79, 82, 95, 96, 97, 98, 118
7	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94
8	99, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112

1993). Then, matrix  $A$  in (2) is given by  $A^T = [A_1^T, \dots, A_I^T]$ , where, for each  $i = 1, \dots, I$ , the block of rows  $A_i$  corresponding to Node  $i$  is formed by stacking the matrices  $B_p$  corresponding to all PMUs contained in Node  $i$ , i.e.,  $A_i^T = [B_{p_1}^T, \dots, B_{p_{p_i}}^T]$ , where  $p_1, \dots, p_{p_i}$  denote the indexes of those PMUs.

We place the PMUs using the method in Tai et al. (2013). This guarantees that matrix  $A$  has full column rank. We also assume that the noise covariance is  $R = \sigma^2 I$ , with  $\sigma = 0.05$ . Notice that voltage and current values in the test system are *per unit* values, i.e., they appear divided by the nominal voltage  $V_0$  and the nominal current  $I_0$ , respectively. Hence,  $\sigma = 0.05$  means that voltage measurements have a standard deviation of  $0.05 \times V_0$  volts, and current measurements have one of  $0.05 \times I_0$  amperes. This leads to a global estimate  $\hat{x}$  having a relative estimation error of

$$e = 20 \log_{10} \frac{\|x - \hat{x}\|}{\|x\|} = -17.45 \text{ dB.} \quad (23)$$

In the simulations below, we use

$$r(t) = 20 \log_{10} \frac{\|\hat{x} - \hat{x}(t)\|}{\|\hat{x}\|}, \quad (24)$$

to measure the relative difference between the global estimate  $\hat{x}$  and the one yielded, at time step  $t$ , by the proposed distributed algorithms.

### 5.1.1. Cyclic network topology

In the first simulation we cluster the buses into eight nodes, as shown in Table 1. From the definition of  $\mathcal{N}_i$ , it follows that  $j \in \mathcal{N}_i$

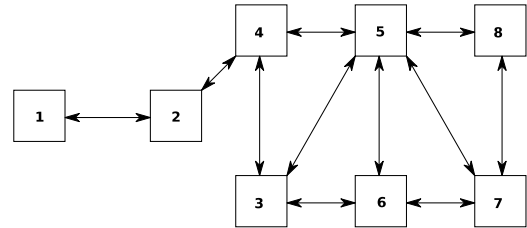


Fig. 2. Cyclic network topology induced by the nodes in Table 1.

if there is a bus in either, Node  $i$  or  $j$ , with a PMU installed, having an attached line coming from a bus inside the other node. Fig. 2 shows the topology of the communication network induced by the clustering given in Table 1.

Fig. 3 shows the convergence of the asymptotic method without preconditioning. To this end, we show the modulus of the estimated voltage of each bus at each step. We see that the convergence is very slow, with a relative difference of  $r(10^6) = -37$  dB, between the global estimate and the one obtained by the distributed algorithm at  $t = 10^6$ . The reason for the slow convergence is that the condition number of  $\Psi$  is 478 972. The preconditioning matrix in (20) gives a condition number of 700, which leads to a much faster convergence. This is shown in Fig. 4, where  $r(2 \times 10^3) = -52.47$  dB. Fig. 5 shows that the convergence of the estimation of  $\|\gamma\|$  and  $\|\gamma^{-1}\|^{-1}$ , at each node, is much faster than that of the WLS estimation algorithm. Finally, Table 2 shows the complexity at each node. To this end, we measure the number of multiplications in a whole cycle of Algorithms 1–3.

**Table 2**

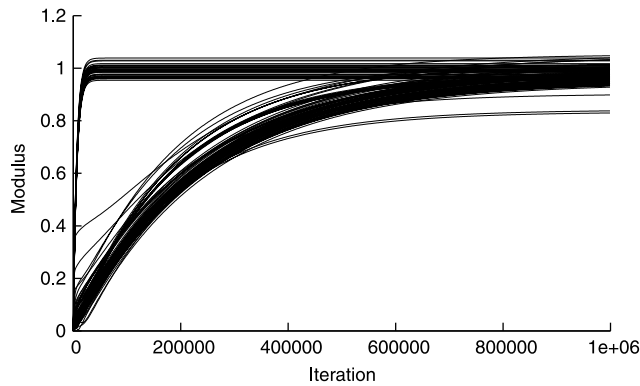
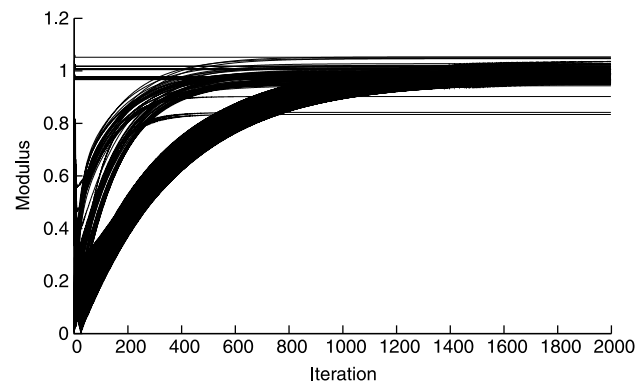
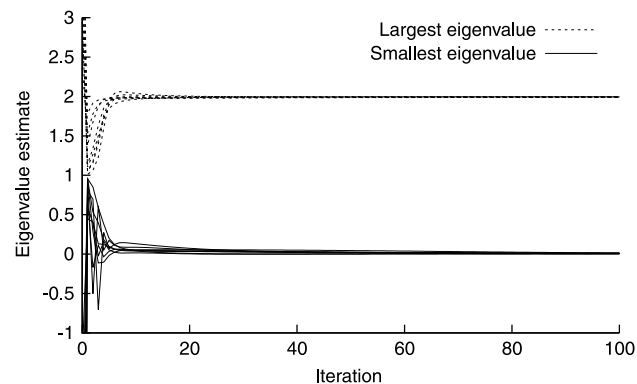
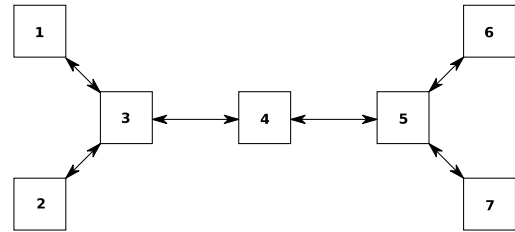
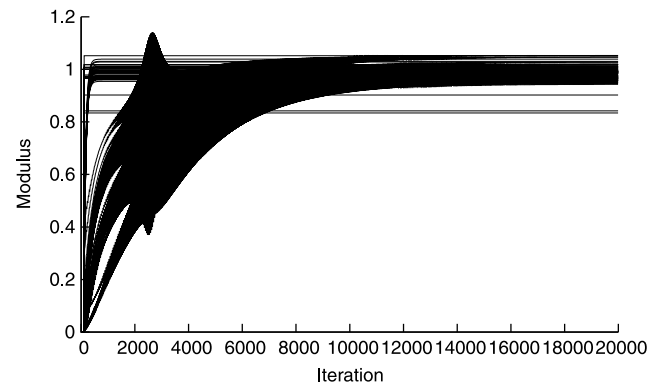
Complexity at each node, in number of multiplications per iteration, for the cyclic network topology.

Node	1	2	3	4	5	6	7	8
Complexity	5202	8613	10 469	14 848	26 054	11 810	14 352	10 628

**Table 3**

Nodes forming an acyclic network topology.

Node	Buses
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 117
2	23, 25, 26, 27, 28, 29, 31, 32, 113, 114, 115
3	5, 16, 17, 18, 19, 20, 21, 22, 24, 30, 33, 34, 35, 36, 37, 39, 40, 71, 72, 73
4	38, 41, 42, 43, 44, 45, 46, 47, 48, 69, 70, 74, 75, 76, 77, 118
5	49, 50, 51, 54, 65, 66, 68, 78, 79, 80, 81, 82, 95, 96, 97, 98, 99, 116
6	52, 53, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 67
7	83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112

**Fig. 3.** Convergence of the asymptotic method, without preconditioning, in a cyclic network.**Fig. 4.** Convergence of the asymptotic method, with preconditioning, in a cyclic network.**Fig. 5.** Convergence of the distributed eigenvalue estimation algorithm.**Fig. 6.** Acyclic network topology induced by the nodes in Table 3.**Fig. 7.** Convergence of the asymptotic method (with preconditioning) in an acyclic network.

### 5.1.2. Acyclic network topology

In the second simulation we do the clustering such that the induced topology is acyclic. From the definition of  $\mathcal{B}_i$ , it follows that  $j \in \mathcal{B}_i$  if there is a bus (possibly neither in Node  $i$  nor in  $j$ ), with a PMU installed, having one neighbor bus (i.e., a bus connected to it via an attached line), including possibly itself, in each node,  $i$  and  $j$ . The clustering and its induced topology are shown in Table 3 and Fig. 6, respectively.

The convergence of the asymptotic method (with preconditioning) is shown in Fig. 7, with a final relative difference with the global estimate of  $r (20 \times 10^3) = -69.67$  dB. In this case, we wait for 100 steps before starting with the distributed estimation. The gap caused by this delay can be seen at the beginning of the graph. We introduced this late start so as to give time for Algorithm 2 and 3 to obtain reasonable approximations of  $\|\mathcal{Y}\|$  and  $\|\mathcal{Y}^{-1}\|^{-1}$ , respectively. We see that the asymptotic method presents an oscillating behavior between time steps 1500 and 3500. This is because the transients in the estimation of the scaling factor  $\gamma(t)$  cause the recursions (8) to become temporarily unstable. We also see that the asymptotic method requires about  $20 \times 10^3$  steps to converge. This is because in this case, preconditioning leads to a



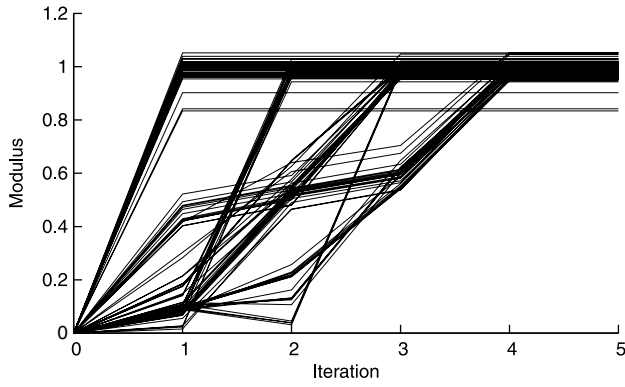


Fig. 8. Convergence of the finite-time method (in an acyclic network).

Table 4

Complexity at each node, in a number of multiplications per iteration, for the acyclic network topology.

Node	Asymptotic method	Finite-time method
1	4985	1912
2	3645	786
3	14088	4400
4	10400	2304
5	17802	3240
6	3891	1267
7	8919	8437

condition number of 5264. On the other hand, the convergence of the finite-time method does not depend on the condition number, but on the network diameter, which in this case is four. Fig. 8 shows the convergence of this method in four steps, with a final error of  $r(4) = -223.7$  dB, caused by numerical inaccuracy.

Table 4 shows the complexity at each node. To this end, we consider that solving the positive-definite linear system for computing  $\check{x}_{i,j}(t)$ , using the Cholesky decomposition, requires  $n^3/3 + 2n^2$  multiplications ( $n^3/3$  for the decomposition and  $2n^2$  for solving two triangular linear systems). Also, computing the inverse of the matrix  $\check{\Sigma}_i(t)$ , using also the Cholesky Decomposition, requires  $n^3/2$  multiplications (Krishnamoorthy & Menon, 2011).

### 5.2. Sensor localization

Sensor localization refers to the problem of obtaining the locations of each node in a network, based on the knowledge of the locations of a few anchor nodes, as well as the mutual distances between neighbor nodes. A distributed method for carrying out this task is proposed in Khan et al. (2009). This method requires that, for each  $i = 1, \dots, I$ , Node  $i$  lies inside of at least one triangle defined by three of its neighbors  $\mathcal{N}_i = \{j, k, l\}$ . Then, the coordinates  $x_i$  of Node  $i$  can be written as

$$x_i = \sum_{j \in \mathcal{N}_i} c_{i,j} x_{i,j}, \quad (25)$$

where the barycentric coordinates  $c_{i,j}$  are given by

$$c_{i,j} = \frac{S(i \cup \mathcal{N}_i \setminus j)}{S(\mathcal{N}_i)},$$

with  $S(i, j, k)$  denoting the area of the triangle formed by Nodes  $i, j$  and  $k$ . The latter can be computed using the Cayley–Menger determinant as follows

$$S^2(i, j, k) = -\frac{1}{16} \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & d_{i,j}^2 & d_{i,k}^2 \\ 1 & d_{i,j}^2 & 0 & d_{j,k}^2 \\ 1 & d_{i,k}^2 & d_{j,k}^2 & 0 \end{vmatrix},$$

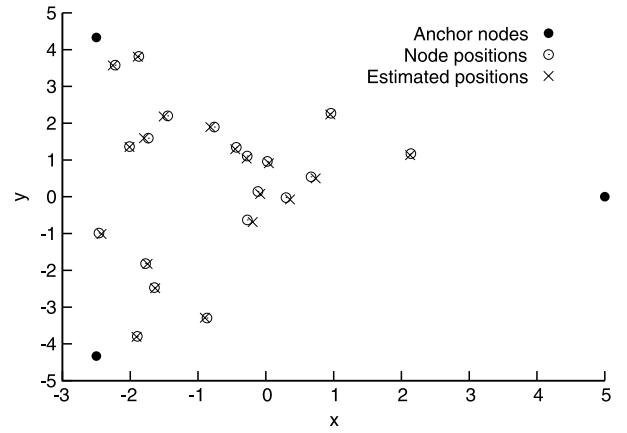


Fig. 9. Node positions and estimates.

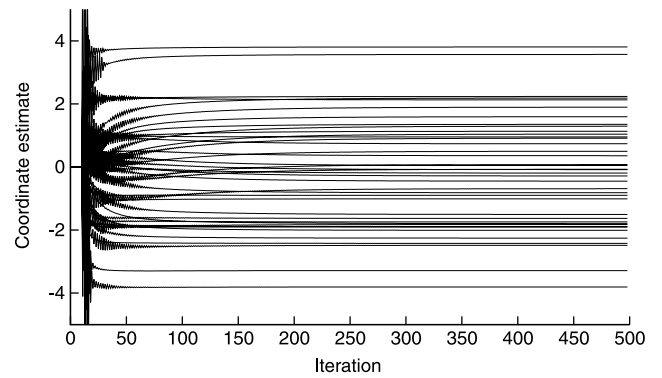


Fig. 10. Convergence of the node coordinate estimates.

where  $d_{i,j} = \|x_i - x_j\|$  denotes the distance between Nodes  $i$  and  $j$ .

For each  $i = 1, \dots, I$ , we have one equation of the form (25), for each triangle containing Node  $i$ . We assume that  $N$  such triangles exist for each node. Hence, we have  $N \times I$  equations. Let  $x_i \in \mathbb{R}^2$ ,  $i = 1, \dots, I$ , denote the node coordinates and  $a_j \in \mathbb{R}^2$ ,  $j = 1, \dots, J$ , denote those of the anchor nodes. Let also  $x^T = [x_1^T, \dots, x_I^T]$  and  $a^T = [a_1^T, \dots, a_J^T]$ . Then, the aforementioned  $N \times I$  equations can be written as

$$x = (C \otimes I_2) x + (D \otimes I_2) a,$$

or equivalently,

$$y = Ax, \quad (26)$$

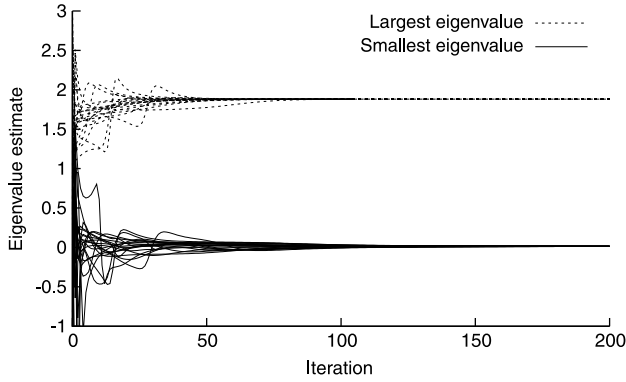
with  $y = (D \otimes I_2) a$  and  $A = I - C \otimes I_2$ . Due to inaccuracy in distance measurements, (26) can be approximately expressed as in (2). In that case, we can use our proposed distributed method to obtain, at each node, a WLS estimation of its coordinates.

The experiment setup is shown in Fig. 9. It includes three anchor nodes, defining a triangle containing  $I = 20$  randomly placed nodes. We use a noise covariance matrix  $R = \sigma^2 I_d$ , where  $I_d$  denotes the identity matrix, and  $\sigma = \sqrt{10^{-3}} \simeq 31.62$  centimeters. With this setup, the global estimate  $\hat{x}$  yields a relative localization error of  $e = -33.39$  dB, defined as in (23). The convergence of the coordinate estimates at each node, using the proposed method, with preconditioning, is shown in Fig. 10. As before, we wait for 10 steps before starting the iterations, to give time for Algorithms 2 and 3 to obtain reasonable approximations of  $\|\gamma\|$  and  $\|\gamma^{-1}\|^{-1}$ , respectively. The convergences of these estimates are shown in Fig. 11. Finally, the complexity at each node is shown in Table 5.

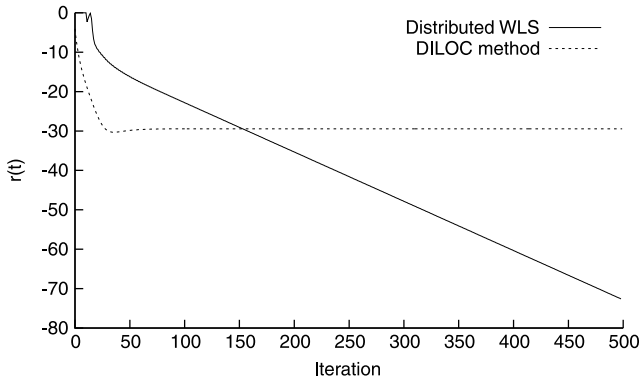
For comparison, we also consider the distributed iterative localization algorithm (DILOC) proposed in Khan et al. (2009). This

**Table 5**  
Complexity at each sensor node, in a number of multiplications per iteration.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Complexity	170	282	282	426	170	282	602	602	282	426	282	426	282	170	170	426	426	426	426	426



**Fig. 11.** Convergence of the estimated eigenvalues.



**Fig. 12.** Relative difference with the global estimate  $\hat{x}$  vs. iteration  $t$ .

method solves (26) using Richardson's recursions to invert matrix  $A$ . This requires that  $N = 1$ , i.e., only one equation of the form (25) is considered for each node. In this case, the recursions are guaranteed to converge because, as the authors show,  $\|I - A\| < 1$  holds in this problem. Fig. 12 shows the evolution of the relative difference  $r(t)$  (defined as in (24)) between the estimates of each method, and the global estimate  $\hat{x}$ . We see that the DILOC method has a faster convergence. This is because the condition number of  $A$  is smaller than that of  $A^*R^{-1}A$ , which is the matrix inverted by our proposed method. However, at  $t = 500$ , the DILOC method yields  $r(500) = -29.44$  dB, while the proposed one gives  $r(500) = -72.71$  dB. This difference results from the fact that the DILOC method does not produce the WLS solution on the limit.<sup>1</sup>

## 6. Conclusion

We proposed two methods for weighted least squares estimation in large-scale systems. Both methods converge to the global solution and aim to maximize the convergence speed. The first method converges asymptotically and involves a distributed estimation of the scaling parameter upon which the convergence

speed depends. To further speed up the convergence, we also use a practically feasible preconditioning method, for which we bounded the speed difference with respect to the fastest theoretically achievable. The second proposed method has an even faster convergence, as it achieves the global optimal in finite time. However, it is only suitable for applications where the graph produced by the communication network contains no loops.

## Appendix A. Proofs of Section 3

### A.1. Proof of Theorem 6

Fix  $t \in \mathbb{N}$ . Let  $k_i(t) \in \mathbb{R}$  and

$$\bar{b}_i(t) = k_i(t)b_i(t). \quad (\text{A.1})$$

From (14), we have

$$\bar{b}_i(t+1) = \frac{k_i(t+1)}{k_i(t)} \Pi_i^{1/2} \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} \frac{k_i(t)}{k_j(t)} \Psi_{i,j}^{(k)} \Pi_j^{1/2} \bar{b}_j(t).$$

Let  $k_i(0) = 1$  and

$$\varsigma_i(t) = \frac{k_i(t)}{k_i(t-1)},$$

so that

$$k_i(t) = \prod_{\tau=1}^t \varsigma_i(\tau).$$

Then,

$$\bar{b}_i(t+1) = \varsigma_i(t+1)\bar{b}_i(t+1), \quad (\text{A.2})$$

with

$$\tilde{b}_i(t+1) = \Pi_i^{1/2} \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} v_{i,j}(t) \Psi_{i,j}^{(k)} \Pi_j^{1/2} \bar{b}_j(t),$$

$$v_{i,j}(t) = \prod_{\tau=1}^t \frac{\varsigma_i(\tau)}{\varsigma_j(\tau)}.$$

We need to design  $k_i(t+1)$ , or equivalently  $\varsigma_i(t+1)$ , to avoid the indefinite increase or decrease of  $b(t)$ . In principle, this could be achieved by choosing

$$k_i(t) = \|b_i(t)\|^{-1},$$

so that  $\|\bar{b}_i(t)\| = 1$ , for all  $t \in \mathbb{N}$ . From (A.2), this would lead to

$$\varsigma_i(t+1) = \|\tilde{b}_i(t+1)\|^{-1}.$$

However, the question then arises as to whether some of the scalars  $v_{i,j}(t)$  would grow to infinity. Notice that

$$v_{i,j}(t) = \frac{k_i(t)}{k_j(t)}.$$

Hence, this could only happen if some vector in the eigenspace associated to the largest eigenvalue of  $\mathcal{Y}$  has zero components in the entries corresponding to  $b_j(t)$ . We call a matrix satisfying this property, *ill-posed*. Although the set of ill-posed matrices is nowhere dense, (i.e., it is unlikely to have an ill-posed matrix  $\mathcal{Y}$ ),

<sup>1</sup> Notice that, in the scenario considered in this work, noisy inter-node distances are only measured once, and they remain unchanged during the whole iteration process. This is in contrast to the scenario considered in Khan et al. (2009), where these distances are re-measured at each iteration.

we can avoid the indefinite growth of  $v_{i,j}(t)$  by choosing  $\varsigma_i(t+1)$  so that  $\|\tilde{b}_i(t+1)\| \leq 1$  and, for all  $j \in \mathcal{B}_i = \{j : \psi_{i,j} \neq 0\}$ ,  $\varsigma_i(t+1)v_{i,j}(t) \leq 1$ .

This leads to

$$\begin{aligned} \varsigma_i(t+1) &= \min \left\{ \left\| \tilde{b}_i(t+1) \right\|^{-1}, v_{i,j}^{-1}(t), j \in \mathcal{B}_i \right\} \\ &= \max \left\{ \left\| \tilde{b}_i(t+1) \right\|, v_{i,j}(t), j \in \mathcal{B}_i \right\}^{-1}. \end{aligned}$$

From (15) and (A.1), the estimate  $\tilde{\gamma}_i(t)$  of  $\|\mathcal{Y}\|$  at  $t$  is

$$\begin{aligned} \tilde{\gamma}_i(t) &= \frac{\|b_i(t)\|}{\|b_i(t-1)\|} \\ &= \varsigma_i^{-1}(t) \frac{\|\tilde{b}_i(t)\|}{\|\tilde{b}_i(t-1)\|}. \end{aligned}$$

However, if  $\mathcal{Y}$  is ill-posed,  $\|\tilde{b}_i(t)\|$  will tend to zero. In such case,  $\tilde{\gamma}_i(t)$  can be computed by

$$\tilde{\gamma}_i(t) = \varsigma_j^{-1}(t) \frac{\|\tilde{b}_j(t)\|}{\|\tilde{b}_j(t-1)\|},$$

for some neighbor node  $j$  for which  $\|\tilde{b}_j(t)\|$  does not tend to zero. Notice that such a neighbor always exists, for otherwise Node  $i$  would be isolated from all other nodes.

### A.2. Proof of Theorem 7

We need use the following result.

**Lemma 16.** Let  $M = [M_{i,j}]_{i,j=1,\dots,l}$  be a block symmetric matrix. Then

$$\|M\| \leq \max_i \psi_i,$$

where

$$\psi_i = \sum_{j=1}^l \|M_{i,j}\|.$$

**Proof.** Let  $y = Mx$ , with  $x = [x_1, \dots, x_l]$  and  $y = [y_1, \dots, y_l]$ . We have

$$\begin{aligned} \|y_i\| &\leq \sum_{j=1}^l \|M_{i,j}\| \|x_j\| \\ &= \sum_{j=1}^l \|M_{i,j}\|^{1/2} \left( \|M_{i,j}\| \|x_j\|^2 \right)^{1/2} \\ &\leq \left( \sum_{j=1}^l \|M_{i,j}\| \right)^{1/2} \left( \sum_{j=1}^l \|M_{i,j}\| \|x_j\|^2 \right)^{1/2}. \end{aligned}$$

Let  $\psi = \max_i \psi_i$ . Then,

$$\begin{aligned} \|y\|^2 &= \sum_{i=1}^l \|y_i\|^2 \\ &\leq \sum_{i=1}^l \left( \sum_{j=1}^l \|M_{i,j}\| \right) \left( \sum_{j=1}^l \|M_{i,j}\| \|x_j\|^2 \right) \\ &\leq \psi \sum_{i=1}^l \sum_{j=1}^l \|M_{i,j}\| \|x_j\|^2 \\ &\leq \psi^2 \sum_{j=1}^l \|x_j\|^2 \\ &= \psi^2 \|x\|^2, \end{aligned}$$

and the result follows.

**Proof of Theorem 7.** Using Lemma 16 we have

$$\|\mathcal{Y}\| \leq \max_i \psi_i,$$

with

$$\begin{aligned} \psi_i &= \sum_{j=1}^l \left\| \Pi_i^{1/2} \Psi_{i,j} \Pi_j^{1/2} \right\| \\ &= \sum_{j=1}^l \left\| \sum_{k:i,j \in \mathcal{O}_k} \Pi_i^{1/2} \Psi_{i,j}^{(k)} \Pi_j^{1/2} \right\| \\ &\leq \sum_{j=1}^l \sum_{k:i,j \in \mathcal{O}_k} \left\| \Pi_i^{1/2} \Psi_{i,j}^{(k)} \Pi_j^{1/2} \right\| \\ &= \sum_{k \in \mathcal{I}_i} \sum_{j \in \mathcal{O}_k} \left\| \Pi_i^{1/2} \Psi_{i,j}^{(k)} \Pi_j^{1/2} \right\|. \end{aligned}$$

Hence,

$$\|\mathcal{Y}\| \leq \max_i \phi_i,$$

and the result follows.

### A.3. Proof of Theorem 8

We need the following lemma.

**Lemma 17.** If  $M = \begin{bmatrix} I & C \\ C^* & I \end{bmatrix} \geq 0$ , then  $\|C\| \leq 1$ .

**Proof.** Let  $u^T = [x^T \ y^T]$ . Then, for any  $u$ ,

$$\begin{aligned} u^* M u &= \|x\|^2 + \|y\|^2 + 2y^T C x \\ &\geq 0. \end{aligned}$$

Choose  $x$  and  $y$  such that  $\|x\| = \|y\| = 1$  and  $y^* C x = -\|C\|$ . Then, the inequality above becomes  $1 - \|C\| \geq 0$ , and the result follows.

**Proof of Theorem 8.** Recall that  $\Pi = \text{diag}\{\Pi_1, \dots, \Pi_l\}$  and let  $D = \tilde{\Pi} \Pi^{-1}$ , with  $\tilde{\Pi} \in \mathcal{P}$ . Then,

$$\tilde{\Pi}^{1/2} \Psi \tilde{\Pi}^{1/2} = D^{1/2} \mathcal{Y} D^{1/2},$$

and we have

$$\begin{aligned} \kappa(\tilde{\Pi}^{1/2} \Psi \tilde{\Pi}^{1/2}) &= \kappa(D^{1/2} \mathcal{Y} D^{1/2}) \\ &= \kappa^2((\mathcal{Y} D)^{1/2}). \end{aligned} \tag{A.3}$$

Let  $\bar{\sigma}(A)$  and  $\underline{\sigma}(A)$  denote the largest and smaller singular values of  $A$ , respectively. Now

$$\begin{aligned} \bar{\sigma}((\mathcal{Y} D)^{1/2}) &= \max_{x \neq 0} \frac{\|(\mathcal{Y} D)^{1/2} x\|}{\|x\|} \\ &= \max_{x \neq 0} \frac{\|\mathcal{Y}^{1/2} x\|}{\|D^{-1/2} x\|} \\ &\geq \frac{\|\mathcal{Y}^{1/2} x_0\|}{\|D^{-1/2} x_0\|}, \end{aligned}$$

for any  $x_0 \neq 0$ . Similarly,

$$\underline{\sigma}((\mathcal{Y} D)^{1/2}) \leq \frac{\|\mathcal{Y}^{1/2} y_0\|}{\|D^{-1/2} y_0\|},$$

for any  $y_0 \neq 0$ . Let  $x_0$  and  $y_0$  have unit norm and be such that  $D^{-1/2}x_0 = \underline{\sigma}(D^{-1/2})x_0$  and  $\Upsilon^{1/2}y_0 = \underline{\sigma}(\Upsilon^{1/2})y_0$ . Then,

$$\begin{aligned} \kappa((\Upsilon D)^{1/2}) &= \frac{\bar{\sigma}((\Upsilon D)^{1/2})}{\underline{\sigma}((\Upsilon D)^{1/2})} \\ &\geq \frac{\|D^{-1/2}y_0\| \|\Upsilon^{1/2}x_0\|}{\underline{\sigma}(D^{-1/2}) \underline{\sigma}(\Upsilon^{1/2})} \\ &\geq \frac{\|\Upsilon^{1/2}x_0\|}{\underline{\sigma}(\Upsilon^{1/2})}. \end{aligned}$$

Now, since  $D^{-1/2}$  is block diagonal,  $x_0$  can be chosen so that its nonzero components correspond to only one block of  $D^{-1/2}$ . Let  $x_{0,b}$  denote the entries of  $x_0$  in that block, and  $\Upsilon_b^{1/2}$  denote the columns of  $\Upsilon^{1/2}$  corresponding to the same block. Then,

$$\begin{aligned} \|\Upsilon^{1/2}x_0\|^2 &= \left| (\Upsilon^{1/2}x_0)^* \Upsilon^{1/2}x_0 \right| \\ &= \left| x_{0,b}^* \Upsilon_b^{1/2} \Upsilon_b^{1/2} x_{0,b} \right| \\ &= \left| x_{0,b}^* x_{0,b} \right| \\ &= \|x_0\|^2 \\ &= 1. \end{aligned}$$

Then,

$$\begin{aligned} \kappa((\Upsilon D)^{1/2}) &\geq \frac{1}{\underline{\sigma}(\Upsilon^{1/2})} \\ &= \frac{\kappa(\Upsilon^{1/2})}{\bar{\sigma}(\Upsilon^{1/2})}. \end{aligned}$$

Let  $\Upsilon = [\Upsilon_{i,j}]_{i,j=1,\dots,l}$  be the block partition of  $\Upsilon$ . From Lemma 17,  $\|\Upsilon_{i,j}\| \leq 1$ , for all  $i, j = 1, \dots, l$ . Then, from Lemmas 16 and 17

$$\begin{aligned} \bar{\sigma}(\Upsilon) &\leq \max_i \sum_{j=1}^l \|\Upsilon_{i,j}\| \\ &\leq \max_i |\mathcal{B}_i| \\ &= \beta. \end{aligned}$$

Hence, from (A.3),

$$\begin{aligned} \kappa(\tilde{\Pi}^{1/2}\Psi\tilde{\Pi}^{1/2}) &\geq \frac{\kappa(\Upsilon)}{\bar{\sigma}(\Upsilon)} \\ &\geq \frac{\kappa(\Upsilon)}{\beta}. \end{aligned}$$

The result follows since the inequality above holds for any  $\tilde{\Pi}$ .

## Appendix B. Proofs of Section 4

**Proof of Lemma 12.** From Kailath, Sayed, and Hassibi (2000, A.1(v)), we have  $\Psi_{a,a} - \Psi_{a,b}\tilde{\Sigma}_b\Psi_{b,a}$  and  $\Psi_{b,b} - \Psi_{b,a}\Psi_{a,a}\Psi_{a,b}$  are invertible, and

$$\Psi^{-1} = \begin{bmatrix} \Sigma_a & -\Sigma_a\Psi_{a,b}\Psi_{b,b}^{-1} \\ -\Sigma_b\Psi_{b,a}\Psi_{a,a}^{-1} & \Sigma_b \end{bmatrix}$$

with  $\Sigma_a = (\Psi_{a,a} - \Psi_{a,b}\Psi_{b,b}^{-1}\Psi_{b,a})^{-1}$  and  $\Sigma_b = (\Psi_{b,b} - \Psi_{b,a}\Psi_{a,a}^{-1}\Psi_{a,b})^{-1}$ . The result then follows from (3).

**Proof of Lemma 13.** Follows immediately by applying Lemma 12 with  $x_a = x_1$  and  $x_b^T = [x_2, \dots, x_l]$ .

Before proving Theorem 15, we introduce some notation.

**Notation 18.** For each  $i \in \mathcal{I} = \{1, \dots, l\}$ , and  $j \in \mathcal{B}_i$ , let  $\mathcal{M}_i(0) = \mathcal{M}_{j,i}(0) = \{i\}$ . Then, for each  $t \in \mathbb{N}$ , define recursively the following two sequences of sets

$$\begin{aligned} \mathcal{M}_i(t) &= \bigcup_{k \in \mathcal{M}_i(t-1)} \mathcal{B}_k, \\ \mathcal{M}_{j,i}(t) &= \bigcup_{k \in \mathcal{M}_{j,i}(t-1)} \mathcal{B}_k \setminus \{j\}. \end{aligned}$$

I.e.,  $\mathcal{M}_i(t)$  is the set of indexes of nodes which are  $t$  edges away from Node  $i$ , and  $\mathcal{M}_{j,i}(t)$  is the set resulting after removing from  $\mathcal{M}_i(t)$  the indexes of those nodes which are linked to Node  $i$  through Node  $j$ . For each  $t \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$  and  $i \in \mathcal{I}$ , let  $\xi_i^T(t) = [x_k^T : k \in \mathcal{M}_i(t)]$ ,  $\xi_{i,j}^T(t) = [x_k^T : k \in \mathcal{M}_{j,i}(t)]$ ,  $\Omega_i(t) = [A_{k,i}]_{k \in \mathcal{I}, i \in \mathcal{M}_i(t)}$  and  $\Omega_{i,j}(t) = [A_{k,i}]_{k \in \mathcal{I}, i \in \mathcal{M}_{j,i}(t)}$ . Also, let  $(\hat{\xi}_i(t), \Xi_i(t))$  be the WLS solution of the reduced system

$$y = \Omega_i(t)\xi_i(t) + v, \quad (\text{B.1})$$

i.e.,

$$\begin{aligned} \hat{\xi}_i(t) &= \Xi_i(t)\Omega_i^*(t)R^{-1}y \\ \Xi_i(t) &= (\Omega_i^*(t)R^{-1}\Omega_i(t))^{-1} \end{aligned}$$

and  $(\hat{\xi}_{i,j}(t), \Xi_{i,j}(t))$  be WLS the solution of

$$y = \Omega_{i,j}(t)\xi_{i,j}(t) + v.$$

**Proof of Theorem 15.** Suppose that, at time  $t \in \mathbb{N}$ , and for each  $i \in \mathcal{I}$  and  $j \in \mathcal{B}_i \setminus \{i\}$ , Node  $i$  is able to compute the components  $(\check{x}_i(t), \check{\Sigma}_i(t))$ , corresponding to the state  $x_i$ , of the solution  $(\hat{\xi}_i(t), \hat{\Xi}_i(t))$ , and the components  $(\check{x}_{i,j}(t), \check{\Sigma}_{i,j}(t))$ , corresponding to the same state, of  $(\hat{\xi}_{i,j}(t), \hat{\Xi}_{i,j}(t))$ . Since the network is acyclic, for each  $i \in \mathcal{I}$  and each  $t \in \mathbb{N}$ , we have

$$\mathcal{M}_i(t+1) = \{i\} \cup \bigcup_{j \in \mathcal{B}_i \setminus \{i\}} \mathcal{M}_{j,i}(t).$$

Then, given that Node  $i$  receives  $\gamma_{i,j}(t)$  and  $\Gamma_{i,j}(t)$ , from each  $j \in \mathcal{B}_i \setminus \{i\}$ ,  $|\mathcal{B}_i| - 1$  applications of Lemma 13 ( $|\mathcal{B}_i|$  denotes the number of elements in the set  $\mathcal{B}_i$ ), gives that  $\Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i\}} \Gamma_{j,i}(t-1)$  is invertible, and Node  $i$  is able to compute  $(\check{x}_i(t+1), \check{\Sigma}_i(t+1))$ . Also,  $|\mathcal{B}_i| - 2$  applications of Lemma 13 give that, for each  $j \in \mathcal{B}_i \setminus \{i\}$ ,  $\Psi_{i,i} - \sum_{j \in \mathcal{B}_i \setminus \{i,j\}} \Gamma_{j,i}(t-1)$  is invertible and Node  $i$  can compute  $(\check{x}_{i,j}(t+1), \check{\Sigma}_{i,j}(t+1))$ . Then, the result follows after initializing the induction above using  $(\check{x}_i(0), \check{\Sigma}_i(0))$ , at each  $i \in \mathcal{I}$ , for which no information exchange is required.

At each  $t \in \mathbb{N}_0$  and  $i \in \mathcal{I}$ ,  $(\check{x}_i(t), \check{\Sigma}_i(t))$  is the WLS solution of the sub-system (B.1). Since (B.1) is obtained by considering only the nodes in  $\mathcal{M}_i(t)$ , and  $\mathcal{M}_i(t) = \mathcal{I}$ , for all  $t \geq \rho_i$ , (22) follows.

## References

- Bertsekas, Dimitri P., & Tsitsiklis, John N. (1997). *Parallel and distributed computation: numerical methods*.
- Carli, R., Chiuso, A., Schenato, L., & Zampieri, S. (2008). Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4), 622–633.
- Christie, Rich (1993). 118 bus power flow test case. [http://www.ee.washington.edu/research/pstca/pf118/pg\\_tca118bus.htm](http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm).
- Conejo, Antonio J., de la Torre, Sebastian, & Canas, Miguel (2007). An optimization approach to multiarea state estimation. *IEEE Transactions on Power Systems*, 22(1), 213–221.

- Dargie, Waltenegus, & Poellabauer, Christian (2010). *Fundamentals of wireless sensor networks: theory and practice* (1st ed.) (p. 8). Wiley.
- Demmel, James (1983). The condition number of equivalence transformations that block diagonalize matrix pencils. *SIAM Journal on Numerical Analysis*, 20(3), 599–610.
- Diao, Yingfei, Fu, Minyue, & Zhang, Huanshui (2013). Localizability and distributed localization of sensor networks using relative position measurements. In *IFAC symposium on large scale systems*.
- Fang, Jun, & Li, Hongbin (2008). Joint dimension assignment and compression for distributed multisensor estimation. *IEEE Signal Processing Letters*, 15, 174–177.
- Gómez-Expósito, Antonio, Villa Jaén, Antonio de la, Gómez-Quiles, Catalina, Rousseaux, Patricia, & Van Cutsem, Thierry (2011). A taxonomy of multi-area state estimation methods. *Electric Power Systems Research*, 81(4), 1060–1069.
- Hlinka, Ondrej, Sluciak, Ondrej, Hlawatsch, Franz, Djuric, Petar M., & Rupp, Markus (2012). Likelihood consensus and its application to distributed particle filtering. *IEEE Transactions on Signal Processing*, 60(8), 4334–4349.
- Huang, Yih-Fang, Werner, Stefan, Huang, Jing, Kashyap, Neelabh, & Gupta, Vijay (2012). State estimation in electric power grids: meeting new challenges presented by the requirements of the future grid. *IEEE Signal Processing Magazine*, 29(5), 33–43.
- Jiang, Weiqing, Vittal, Vijay, & Heydt, Gerald T. (2008). Diakoptic state estimation using phasor measurement units. *IEEE Transactions on Power Systems*, 23(4), 1580–1589.
- Kailath, Thomas, Sayed, Ali H., & Hassibi, Babak (2000). *Linear estimation*. Prentice Hall.
- Kar, Soumyya, Moura, José M. F., & Ramanan, Kavita (2012). Distributed parameter estimation in sensor networks: nonlinear observation models and imperfect communication. *IEEE Transactions on Information Theory*, 58(6), 3575–3605.
- Kay, Steven (1993). *Fundamentals of statistical signal processing, volume 1: estimation theory. Vol. 1* (1st ed.) (p. 4). Prentice Hall.
- Khan, Usman A., Kar, Soumyya, & Moura, José M. F. (2009). Distributed sensor localization in random environments using minimal number of anchor nodes. *IEEE Transactions on Signal Processing*, 57(5), 2000–2016.
- Khan, Usman A., & Moura, José M. F. (2008). Distributing the kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56(10), 4919–4935.
- Krishnamoorthy, Aravindh, & Menon, Deepak (2011). Matrix inversion using cholesky decomposition. arXiv Preprint arXiv:1111.4144.
- Li, Junlin, & AlRegib, Ghassan (2007). Rate-constrained distributed estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55(5), 1634–1643.
- Li, Junlin, & AlRegib, Ghassan (2009). Distributed estimation in energy-constrained wireless sensor networks. *IEEE Transactions on Signal Processing*, 57(10), 3746–3758.
- Lopes, Cassio G., & Sayed, Ali H. (2008). Diffusion least-mean squares over adaptive networks: formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7), 3122–3136.
- Marelli, Damián, & Fu, Minyue (2013). Distributed weighted least squares estimation with fast convergence in large-scale systems. In *IEEE conference on decision and control (CDC)* (pp. 5432–5437).
- Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.
- Ribeiro, Alejandro, & Giannakis, Georgios B. (2006a). Bandwidth-constrained distributed estimation for wireless sensor networks—part I: Gaussian case. *IEEE Transactions on Signal Processing*, 54(3), 1131–1143.
- Ribeiro, Alejandro, & Giannakis, Georgios B. (2006b). Bandwidth-constrained distributed estimation for wireless sensor networks—part II: unknown probability density function. *IEEE Transactions on Signal Processing*, 54(7), 2784–2796.
- Ribeiro, Alejandro, Schizas, I., Roulmliotis, S., & Giannakis, G. (2010). Kalman filtering in wireless sensor networks. *IEEE Control Systems*, 30(2), 66–86.
- Tai, Xin, Marelli, Damián, Rohr, Eduardo, & Fu, Minyue (2013). Optimal PMU placement for power system state estimation with random component outages. *International Journal of Electrical Power & Energy Systems*, 51, 35–42.
- Xiao, Jin-Jun, Ribeiro, Alejandro, Luo, Zhi-Quan, & Giannakis, Georgios B. (2006). Distributed compression-estimation using wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4), 27–41.
- Yang, Peng, Freeman, Randy A., Gordon, Geoffrey J., Lynch, Kevin M., Srinivasa, Siddhartha S., & Sukthankar, Rahul (2010). Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2), 390–396.



**Damián Edgardo Marelli** received his Bachelors Degree in Electronics Engineering from the Universidad Nacional de Rosario, Argentina, in 1995, a Ph.D. degree in Electrical Engineering and a Bachelor (Honours) degree in Mathematics from the University of Newcastle, Australia, in 2003. From 2004 to 2005 he held a postdoctoral position at the Laboratoire d'Analyse Topologie et Probabilités, CNRS/Université de Provence, France. Since 2006 he is Research Academic at the School of Electrical Engineering and Computer Science at the University of Newcastle, Australia. In 2007 he received a Marie Curie Postdoctoral Fellowship, hosted at the Faculty of Mathematics, University of Vienna, Austria, and in 2010 he received a Lise Meitner Senior Fellowship, hosted at the Acoustics Research Institute of the Austrian Academy of Sciences. His main research interests include signal processing and communications.



**Minyue Fu** received his Bachelor's Degree in Electrical Engineering from the University of Science and Technology of China, Hefei, China, in 1982, and M.S. and Ph.D. degrees in Electrical Engineering from the University of Wisconsin-Madison in 1983 and 1987, respectively. From 1983 to 1987, he held a teaching assistantship and a research assistantship at the University of Wisconsin-Madison. He worked as a Computer Engineering Consultant at Nicolet Instruments, Inc., Madison, Wisconsin, during 1987. From 1987 to 1989, he served as an Assistant Professor in the Department of Electrical and Computer Engineering, Wayne State University, Detroit, Michigan. He joined the Department of Electrical and Computer Engineering, the University of Newcastle, Australia, in 1989. Currently, he is a Chair Professor in Electrical Engineering and Head of School of Electrical Engineering and Computer Science. In addition, he was a Visiting Associate Professor at University of Iowa in 1995–1996, and a Senior Fellow/Visiting Professor at Nanyang Technological University, Singapore, 2002. He holds a Qian-ren Professorship at Zhejiang University, China. He is a Fellow of IEEE. His main research interests include control systems, signal processing and communications. He has been an Associate Editor for the *IEEE Transactions on Automatic Control*, *Automatica* and *Journal of Optimization and Engineering*.