Convergence and Accuracy Analysis for a Distributed Static State Estimator based on Gaussian Belief Propagation

Damian Marelli, Tianju Sui†, Minyue Fu, *Fellow IEEE*, and Ximing Sun

***Abstract*—This paper focuses on the distributed static estimation problem. A Belief Propagation (BP) based estimation algorithm is studied for its convergence and accuracy. More precisely, we give conditions under which the BP based distributed estimator is guaranteed to converge and we give concrete characterizations for its accuracy. Our results reveal new insights and properties of this distributed algorithm, leading to better theoretical understanding of static distributed state estimation and new applications of the algorithm.**

***Index Terms*—Distributed State Estimation, Belief Propagation, Convergence Analysis, Accuracy Analysis.**

I. Introduction

Large-scale systems, such as power grid, sensor monitoring networks and telecommunication systems, are receiving increasing attention in different fields. As the data size in the system increases rapidly, the classical centralized estimation approach is no longer suitable because it is often impractical to congregate all the information in a central computing processor and it is unnecessary for each node to estimate the global state. Instead, the distributed estimation approach is required where every node in the network estimates its own *state* using its own measurements and information shared from its neighbouring nodes, and this is done in an iterative fashion.

An example of distributed estimation is the sensor network localization problem where each node (sensor) needs to estimate its own location using relative measurements (e.g., relative distances or relative positions) between sensors and the absolute locations of some (a few) anchor nodes [1],[2]. In this case, it is unnecessary for each sensor to localize other nodes. A distributed method is preferred so that estimation is done in each node, rather than relying on a central processor [3],[4].

For large-scale systems, the goal for static state estimation is that the composite estimate of the whole system, consisting of all local estimates, should approach the optimal estimate obtained by a centralized estimation method using all the measurements. The technical difficulty for a large-scale system is much higher than that for a small-scale one. Several approaches are available in the literature for solving the distributed weighted least-squares (WLS) problem. In [5],[6], each node runs a local estimator for the composite state and a consensus based algorithm is used to achieve the averaging of local information vectors and information matrices (which are related to state estimates and covariances, respectively), resulting in the asymptotic convergence of the local estimates. In [3], a gradient based algorithm (see Step 4 of Algorithm 2 in [3]) is used to solve the distributed WLS problem. In [7], a novel algorithm for distributed WLS is developed based on the Richardson method for solving linear equations. However, these algorithms all have relatively slow convergence. In [8], a project based algorithm is proposed to solve the distributed WLS problem. This algorithm can converge in a finite number of iterations, but the computational complexity for each node in each iteration is proportional to the network size, thus not suitable for large-scale systems.

In this paper, we study a distributed algorithm for static state estimation suitable for large-scale systems. This algorithm is based on the celebrated *Pearl's Belief Propagation* (or *Belief Propagation* (BP) for short) algorithm for statistical learning. Originally proposed by Pearl [9] in 1982, BP (also known as *sum-product message passing*), is a *message passing* algorithm for computing marginal probability density functions (PDFs) on Bayesian networks (directed and acyclic graphs) and Markov random fields (undirected and cyclic graphs). Note that the static state estimation problem can be formulated as the marginal PDF computation problem (more on this later). It is well known that, for acyclic graphs (graphs without loops), the BP successfully computes marginal PDFs in a finite number of iterations. The knowledge that the BP algorithm outperforms other distributed iterative methods can be traced back to the invention of the BP algorithm [9], and has been supported by lots of empirical evidences. The best example is the use of the algorithm in turbo decoding, which makes it possible for practical communications systems to approach the theoretical barrier of Shannon Channel Capacity! (See [10] and references therein.)

However, an unsolved fundamental question is that, for a cyclic network graph, under what conditions will BP iterations converge? For a general cyclic graph, [11], [12], [13], [14], [15] studied the convergence condition for BP. However, these references only gave partial answers. Explicit comparisons with Jacobi and Richardson iterations can be found in [7] and [16]. In [7], we compared a distributed algorithm based on

Damian Marelli is with Guangdong University of Technology, Guangdong Key Laboratory of IoT Information Technology, and French-Argentine International Center for Information and Systems Sciences (email: Damian.Marelli@newcastle.edu.au). Tianju Sui and Ximing Sun are with Dalian University of Technology, China (email:suitj@dlut.edu.cn; sunxm@dlut.edu.cn). Minyue Fu is with University of Newcastle and Guangdong University of Technology (email: minyue.fu@newcastle.edu.au).

†Corresponding author.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61803068, 61633014, 61803101 and U1701264) and the Fundamental Research Funds for the Central Universities (DUT19RC(3)027).

Richardson method and a distributed algorithm based on Gaussian BP, and the advantage of the latter was explained. Jacobi iterations apply to systems with diagonal dominance. In [16], Gaussian BP was studied for generalised diagonally dominant systems by using the so-called walk summability approach to draw the connection between Gaussian BP iterations and Jacobi iterations (which are linear iterations). The advantage of Gaussian BP is implied by the fact that Gaussian BP iterations absorb a lot more "walks" than the Jacobi iterations do for the same iteration number, resulting in much faster calculation.

The distributed algorithm we study is the Gaussian BP algorithm [17], a variant of BP algorithm specialized to Gaussian distributions. The algorithm computes iteratively the mean and variance (or covariance) of each marginal. Several conditions ensuring the convergence of the marginal means and variances have been proposed [17], [16], [18], [19]. But several major drawbacks exist. Firstly, the convergence conditions are too difficult to check. For examples, the convergence condition for the mean in [19] requires the evaluation of the spectral radius of an infinite dimensional matrix. Secondly, the convergence conditions are too strict. For examples, the work in [17] requires a *strict diagonal dominance* condition for measurements and [20] extends [17] by relaxing the condition to *generalised diagonal dominance*. Thirdly, characterization of the convergence rate is difficult. For example, the convergence rate description in [21] is not explicit. Finally, most of these convergence analysis are done only for scalar subsystems (i.e., the state variable of each node is a scalar). Since the state components for each node are not independent in general, results for scalar variables are not applicable to vector variables. The only noticeable exception is perhaps the work of [22] which generalizes the convergence conditions of [19]. But as in [19], the convergence analysis relies on the evaluation of the spectral radius of an infinite dimensional matrix, limiting the results to theoretical interests only.

In this paper, viewing the iterations of the distributed static state estimation algorithm as a dynamic process, we provide conditions under which its stability (i.e., the convergence of the distributed state estimator) is guaranteed. We focus on the networks for which each node has a vector variable and we want to determine the convergence rates of the state estimate and covariance for each node. In contrast to [19] and [22], we provide conditions under which we can qualify the estimation error generated by the distributed algorithm for each iteration number. In particular, our estimation error and covariance formulas clearly explain the impact of the so-called *cycle-free depth* of each node to the estimation accuracy.

The significance of our work lies in both theoretical contributions and potential applications. Firstly, convergence and accuracy analysis for the distributed static state estimator is theoretically meaningful. Secondly, the distributed estimation algorithm is shown to have very good performance on large-scale systems with large *cycle-free depth*.

The rest of this paper is organized as follows. In Section III, the problem formulation and distributed state estimator are given. Section IV studies the convergence of the information matrices. Section V investigates the convergence of the state estimates. The accuracy of the information matrices and state estimates are analyzed in Section VI and VII, respectively. Concluding remarks are stated in Section VIII. Most of the proofs are moved to the full version [23].

II. Preliminary for Belief Propagation

The BP algorithm [9] concerns with a system represented by a *bipartite graph* with I *variable nodes* and V *factor nodes*, as depicted in the Fig. 1. Each variable node i is associated with a random vector $x_i \in \mathbf{R}^{n_i}$ and each factor node v is connected to a subset of variable nodes, $\mathcal{F}_v \subset \{1, 2, \ldots, I\}$. Denoting the joint (or global) variable by $X = \{x_i : i = 1, 2, \ldots, I\}$, it is assumed that its joint PDF $f(X)$ can be expressed in a factor form:

$$f(X) = \prod_{v=1}^{V} f_v\left(X_v\right),$$

where $X_v = \{x_i : i \in \mathcal{F}_v\}, v = 1, 2, \ldots, V$. Each $f_v(X_v)$ represents a piece of partial "knowledge" about X.

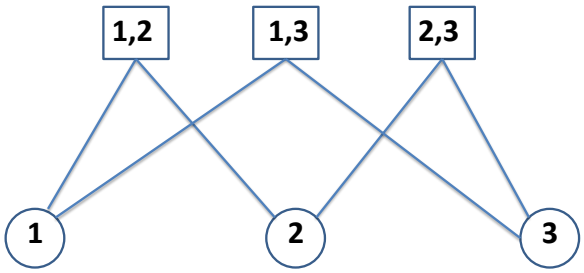


Fig. 1. Bipartite graph: circles = variable nodes; squares = factor nodes

The goal of BP is to compute, at each node i, the marginal $g_i(x_i)$ of $f(X)$, which is defined by

$$g_i(x_i) = \int f(X) d(X \setminus x_i), \tag{1}$$

where $X \setminus x_i$ is the set obtained from X by removing x_i. The algorithm resorts to iterative computation and local communication between connected variable nodes and factor nodes. More specifically, the algorithm starts by each factor node v sending to each variable node $i \in \mathcal{F}_v$ the following marginal PDF (called *message*)

$$m_{v\to i}^{(0)}(x_i) = \int f_v(X_v) d(X_v \setminus x_i). \tag{2}$$

Then, at each iteration $k = 1, 2, \ldots$, each variable node i sends to every connected factor node v the following message:

$$m_{i\to v}^{(k)}(x_i) = \prod_{w \in \mathcal{N}_i \setminus v} m_{w\to i}^{(k)}(x_i), \tag{3}$$

where $\mathcal{N}_i$ is the set of factor nodes connected to variable node i. Similarly, each factor node v sends to every connected variable node i the following message:

$$m_{v\to i}^{(k)}(x_i) = \int f_v(X_v) \prod_{j \in \mathcal{F}_v \setminus i} m_{j\to v}^{(k-1)}(x_j) d(X_v \setminus x_i). \tag{4}$$

The desired marginal at node i and iteration N is estimated:

$$g_i^{(N)}(x_i) = \prod_{w \in \mathcal{N}_i} m_{w\to i}^{(N)}(x_i), \tag{5}$$

modulo a constant scalar to make its integral equal 1.

It is clear that this is a *fully distributed algorithm* because only local information gets exchanged and used without the need for any global information. Note that BP is also known as the *sum-product* algorithm because (2)-(4) use sums (i.e., integrals) and products.

III. Problem Formulation

Consider a system with I unknown local states $x_1, x_2, \ldots, x_I$ and each of them corresponds to a sensing node. Since we focus on the static estimation problem, the local states are unknown time-invariant vectors. Following the notations in BP, we call the sensing nodes *variable nodes* and $x_i \in \mathbb{R}^{n_i}$ is a vector. Associated with the system are two kinds of measurements (also vectors), the so-called *self measurement* for node i,

$$z_i = C_i x_i + v_i, \tag{6}$$

and (pair-wise) *Joint measurement* between nodes i and j,

$$z_{i,j} = C_{i,j} x_i + C_{j,i} x_j + v_{i,j}. \tag{7}$$

In the above, the matrices $C_i, C_{i,j}$ and $C_{j,i}$ are known; v_i and $v_{i,j}$ are independent Gaussian measurement noises with known covariances $R_i > 0$ and $R_{i,j} > 0$, respectively. Note that: 1) the factor node (i,j) is unordered, i.e., $(i,j) = (j,i)$; 2) $z_{i,j} = z_{j,i}$ and $v_{i,j} = v_{j,i}$; 3) It is not necessary for all variable nodes to have self measurements or all variable node pairs to have joint measurements. In fact, joint measurements are typically sparse for large graphs.

The problem of distributed WLS estimation is to compute the maximum likelihood (ML) estimate for each x_i and the corresponding estimation error covariance using a fully distributed algorithm. It is clear that the likelihood functions given by the self and joint measurements are, respectively,

$$\begin{aligned} f_i(x_i) &= p\left(z_i | x_i\right) \\ &\sim \mathcal{N}(z_i - C_i x_i, R_i), \end{aligned} \tag{8}$$

$$\begin{aligned} f_{i,j}(x_i, x_j) &= p\left(z_{i,j} | x_i, x_j\right) \\ &\sim \mathcal{N}(z_{i,j} - C_{i,j} x_i - C_{j,i} x_j, R_{ij}), \end{aligned} \tag{9}$$

where $\mathcal{N}(\mu, \Sigma)$ stands for a Gaussian PDF with mean μ and covariance Σ. It is noted that, in our setting, $f_i(x_i)$ in (8) corresponds to the variable node and $f_{i,j}(x_i, x_j)$ in (9) corresponds to the factor node.

The joint likelihood function for $X = \{x_i : i = 1, 2, \ldots, I\}$ becomes

$$f(X) = \prod_i f_i(x_i) \prod_{(i,j)} f_{i,j}(x_i, x_j). \tag{10}$$

Therefore, the maximum likelihood function for each x_i is given by

$$g_i(x_i) = \int f(X) d(X \setminus x_i), \tag{11}$$

which is exactly the task of static distributed state estimator.

The Gaussian BP algorithm computes the mean and covariance for each $g_i(x_i)$ iteratively. That is, for each iteration k, each node i computes an estimate of the mean, $\hat{x}_i(k)$, and an estimate of the covariance $\Sigma_i(k)$. The detailed algorithm and its derivation can be found in [17]. In this paper, we simply rewrite the Gaussian BP algorithm for the measurements (6)-(7). This is listed in Algorithm 1.

Two variables in Algorithm 1 are of particular importance:

$$\alpha_i(N) = Q_i(N) \hat{x}_i(N); \quad Q_i(N) = \Sigma_i^{-1}(N), \tag{12}$$

which we call *information vector* and *information matrix*, or *information parameters* collectively.

The convergence and accuracy analysis in our paper requires the following key assumption.

Assumption 1. *For all $i = 1, 2, \ldots, I$ and $j \in \mathcal{N}_i$,*

$$\Omega_{i,j} = C_i^T R_i^{-1} C_i + \sum_{w \in \mathcal{N}_i \setminus j} C_{i,w}^T R_{i,w}^{-1} C_{i,w} > C_{i,j}^T R_{i,j}^{-1} C_{i,j}. \tag{13}$$

Remark 1. Roughly speaking, (13) means that, for each node i, the information contribution from any single neighbouring node j (i.e., $C_{i,j}^T R_{i,j}^{-1} C_{i,j}$) is strictly smaller than the sum of that from node i (i.e., $C_i^T R_i^{-1} C_i$) and all other neighbouring nodes $w \in \mathcal{N}_i \setminus j$ (i.e., $C_{i,w}^T R_{i,w}^{-1} C_{i,w}$). In particular, Assumption 1 implies that $\Omega_{i,j} > 0$ for all (i,j). It also implies that, for every leaf node[1] i, $C_i^T R_i^{-1} C_i > 0$ (or equivalently, C_i has full column rank), due to the fact that the sum term in $\Omega_{i,j}$ is void in this case.

Remark 2. Due to the strict inequality above, it is clear that Assumption 1 is equivalent to the existence of some constant $0 < \eta < 1$ such that

$$\eta \Omega_{i,j} \geq C_{i,j}^T R_{i,j}^{-1} C_{i,j} \tag{14}$$

for all $j \in \mathcal{N}_i$. We will use this property in the sequel.

To a given set of variable and factor nodes, we associate an undirected graph, called the *canonical graph*, which we denote by $\mathcal{G}$. This graph has a node associated with each variable node $i = 1, \ldots, I$, and an edge between nodes i and j, if there exists joint measurement $z_{i,j}$, i.e., for all $j \in \mathcal{N}_i$. Moreover, the edge (i,j) is unordered and we also call it factor node (i,j).

It is well known that the Algorithm 1 converges to the correct marginals in a finite number of iterations when $\mathcal{G}$ is acyclic [24]. In fact, the required number of iterations equals to the diameter of the graph, i.e., the maximum distance of any pair of variable nodes, where the distance of two nodes is the number of edges of the shortest path between them. The fundamental challenge in this paper is to understand how the algorithm performs for cyclic graphs. As mentioned in Section I, the goal of this paper is of twofold: First, we want to provide conditions to guarantee the convergence of Algorithm 1 when the induced bipartite graph is cyclic. Secondly, when convergence occurs, we want to quantify the accuracy of the distributed state estimate, i.e., the difference between our state estimate and the true (or global) maximum likelihood estimate in (11).

[1] A variable node is called a *leaf node* if it is connected by only one edge.

Algorithm 1 A BP-based Distributed Static State Estimator

1) **Initialization:** At time $k=0$, factor node (i,j) sends to each connected variable node i:

$$\alpha_{i,j\to i}(0)=C_{i,j}^T R_{i,j}^{-1} z_{i,j},$$
$$Q_{i,j\to i}(0)=C_{i,j}^T R_{i,j}^{-1} C_{i,j}. \tag{15}$$

2) **Main loop:** At time $k=1,2,\cdots$, do:

2.1) Each variable node i computes

$$\alpha_i(k)=C_i^T R_i^{-1} z_i+\sum_{j\in\mathcal{N}_i}\alpha_{i,j\to i}(k-1),$$
$$Q_i(k)=C_i^T R_i^{-1} C_i+\sum_{j\in\mathcal{N}_i} Q_{i,j\to i}(k-1), \tag{16}$$

and (if required at this iteration)

$$\hat{x}_i(k)=Q_i^{-1}(k)\alpha_i(k),$$
$$\Sigma_i(k)=Q_i^{-1}(k). \tag{17}$$

2.2) Each variable node i sends to each factor node (i,j) with $j\in\mathcal{N}_i$:

$$\alpha_{i\to i,j}(k)=\alpha_i(k)-\alpha_{i,j\to i}(k-1),$$
$$Q_{i\to i,j}(k)=Q_i(k)-Q_{i,j\to i}(k-1), \tag{18}$$

2.3) Each factor node (i,j) sends to each variable node j with $j\in\mathcal{N}_i$:

$$\alpha_{i,j\to j}(k)=C_{j,i}^T R_{i,j\to j}^{-1}(k) z_{i,j\to j}(k),$$
$$Q_{i,j\to j}(k)=C_{j,i}^T R_{i,j\to j}^{-1}(k) C_{j,i}, \tag{19}$$

where

$$z_{i,j\to j}(k)=z_{i,j}-C_{i,j}Q_{i\to i,j}^{-1}(k)\alpha_{i\to i,j}(k),$$
$$R_{i,j\to j}(k)=R_{i,j}+C_{i,j}Q_{i\to i,j}^{-1}(k)C_{i,j}^T. \tag{20}$$

IV. Convergence Analysis for Information Matrices

In this section, we provide our first key result which shows that the information matrices $Q_i(k)$ always converge exponentially to a positive definite matrix, under Assumption 1. In addition, the rate of convergence is also characterized.

Firstly, some preliminary lemmas are required.

Lemma 1. *For any $k\in\mathbb{N}$, $1\le i\le I$ and $j\in\mathcal{N}_i$,*

$$Q_{i\to i,j}(k+1)\le Q_{i\to i,j}(k);$$
$$Q_{i,j\to j}(k+1)\le Q_{i,j\to j}(k);$$
$$R_{i,j\to j}(k+1)\ge R_{i,j\to j}(k). \tag{21}$$

In particular, $Q_{i\to i,j}(k)\le\Omega_{i,j}$ for all $k\ge 1$.

Proof. Given in Lemma 1 of the full version [23]. □

Lemma 2. *Under Assumption 1, for every $1\le i\le I$ and $j\in\mathcal{N}_i$, we have*

$$Q_{i\to i,j}(\infty)=\lim_{k\to\infty}Q_{i\to i,j}(k)>0;$$
$$R_{i,j\to j}(\infty)=\lim_{k\to\infty}R_{i,j\to j}(k)<\infty.$$

Proof. Given in Lemma 2 of the full version [23]. □

Next, we give the main result on convergence. Define

$$\Delta Q_{i\to i,j}(k)=Q_{i\to i,j}^{-1/2}(\infty)Q_{i\to i,j}(k)Q_{i\to i,j}^{-1/2}(\infty)-I;$$
$$\Delta Q_{i,j\to j}(k)=Q_{i,j\to j}^{-1/2}(\infty)Q_{i,j\to j}(k)Q_{i,j\to j}^{-1/2}(\infty)-I;$$
$$\Delta R_{i,j\to j}(k)=R_{i,j\to j}^{-1/2}(\infty)R_{i,j\to j}(k)R_{i,j\to j}^{-1/2}(\infty)-I.$$

Also, let constants $\rho>0$ and $\alpha>0$ be defined as follows:

$$\rho=\max_{i,j}\|R_{i,j\to j}^{-1/2}(\infty)C_{i,j}Q_{i\to i,j}^{-1}(\infty)C_{i,j}^T R_{i,j\to j}^{-1/2}(\infty)\|, \tag{22}$$

$$\alpha=\max_{i,j}\|Q_{i\to i,j}^{-1/2}(\infty)\Omega_{i,j}Q_{i\to i,j}^{-1/2}(\infty)-I\|. \tag{23}$$

Note that $\rho<1$ follows from $R_{i,j}>0$ and

$$R_{i,j\to j}(\infty)=R_{i,j}+C_{i,j}Q_{i\to i,j}^{-1}(\infty)C_{i,j}^T.$$

Lemma 3. *Under Assumption 1, for every node i, its neighbor node j and all $k\in\mathbb{N}$, we have*

$$0\le\Delta Q_{i\to i,j}(k)\le\alpha\rho^{k-1}I. \tag{24}$$

Proof. Given in Lemma 3 of the full version [23]. □

Since ultimately we are only interested in the information matrices $Q_i(k)$, we get the following result from Lemma 3.

Theorem 1. *Under Assumption 1, it holds that $Q_i(k)\to Q_i(\infty)>0$ as $k\to\infty$, for every node i of $\mathcal{G}$. Moreover, by defining*

$$\Delta Q_i(k)=Q_i^{-1/2}(\infty)Q_i(k)Q_i^{-1/2}(\infty)-I,$$

it holds, for every node i of $\mathcal{G}$ and all $k\in\mathbb{N}$, that

$$0\le\Delta Q_i(k)\le\alpha\rho^{k-1}I, \tag{25}$$

where ρ and α are defined in (22) *and* (23)*, respectively. As analyzed before, we have $\rho<1$.*

Proof. Given in Theorem 1 of the full version [23]. □

Remark 3. The result in Theorem 1 shows that the information matrix(i.e., the inverse of covariance matrix) from Algorithm 1 exponentially converges under Assumption 1, and the convergence rate is $\rho<1$.

V. Convergence Analysis for the Estimates

In this section, we proceed to study the convergence of the estimates $\hat{x}_i(k)$. Under Assumption 1, we establish a necessary and sufficient condition for the asymptotic convergence of the estimates. This result is general and non-conservative but requires checking the stability of a high-dimensional matrix. We then provide a sufficient condition for convergence of the estimates which can be easily verified in a distributed fashion, with low computational complexity. As a by-product, we also provide an alternative proof for the known result that the estimates always converge for graphs with at most a single cycle [25].

From (16) and (18), we get

$$\alpha_{i\to i,j}(k+1)=C_i^T R_i^{-1} z_i+\sum_{w\in\mathcal{N}_i\setminus j}\alpha_{i,w\to i}(k).$$

Similarly, from (19) and (20), we get

$$\begin{aligned}\alpha_{i,w\to i}(k) =& C_{i,w}^T R_{i,w\to i}^{-1}(k) z_{w,i} \\ & - C_{i,w}^T R_{i,w\to i}^{-1}(k) C_{w,i} Q_{w\to i,w}^{-1}(k) \alpha_{w\to i,w}(k).\end{aligned}$$

Combining the above two equations gives the following dynamics:

$$\begin{aligned}&\alpha_{i\to i,j}(k+1) \\ =&\beta_{i\to i,j}(k) \\ &- \sum_{w\in\mathcal{N}_i\backslash j} C_{i,w}^T R_{i,w\to i}^{-1}(k) C_{w,i} Q_{w\to i,w}^{-1}(k) \alpha_{w\to i,w}(k),\end{aligned}$$

where

$$\beta_{i\to i,j}(k) = C_i^T R_i^{-1} z_i + \sum_{w\in\mathcal{N}_i\backslash j} C_{i,w}^T R_{i,w\to i}^{-1}(k) z_{w,i}. \quad (26)$$

It is easy to check with Algorithm 1 that the above holds for all $k \geq 1$, provided that, in the equation above, we initialize all $\alpha_{i\to i,j}(0) = 0$.

Defining

$$\begin{aligned}\tilde{x}_{i\to i,j}(k) &= Q_{i\to i,j}^{-1/2}(k) \alpha_{i\to i,j}(k), \\ b_{i\to i,j}(k) &= Q_{i\to i,j}^{-1/2}(k+1) \beta_{i\to i,j}(k), \\ a_{i\to i,j}(k) &= C_{j,i}^T R_{i,j\to j}^{-1}(k) C_{i,j} Q_{i\to i,j}^{-1/2}(k),\end{aligned}$$

we get the following alternative dynamics:

$$\begin{aligned}\tilde{x}_{i\to i,j}(k+1) =& b_{i\to i,j}(k) - Q_{i\to i,j}^{-1/2}(k+1) \\ & \cdot \sum_{w\in\mathcal{N}_i\backslash j} a_{w\to i,w}(k) \tilde{x}_{w\to i,w}(k).\end{aligned}$$

Let S be any ordered sequence of all $(i \to i,j)$. Form the column vector $\tilde{x}(k)$ by stacking up all the $\tilde{x}_{i\to i,j}(k)$ according to S, and similarly form $b(k)$ by stacking up all the $b_{i\to i,j}(k)$.

For each $(i \to i,j)$, define the row vector $A_{i\to i,j}(k)$ with its $(w \to i,w)$-th element equal to $-Q_{i\to i,j}^{-1/2}(k+1) a_{w\to i,w}(k)$ for each $w \in \mathcal{N}_i\backslash j$, and all other elements zero. That is,

$$\begin{aligned}&A_{i\to i,j}(k)\tilde{x}(k) \\ =& - Q_{i\to i,j}^{-1/2}(k+1) \sum_{w\in\mathcal{N}_i\backslash j} a_{w\to i,w}(k) \tilde{x}_{w\to i,w}(k).\end{aligned}$$

Then we have the following dynamics for $\tilde{x}(k)$:

$$\tilde{x}(k+1) = A(k)\tilde{x}(k) + b(k), \quad (27)$$

where $A(k)$ is a matrix formed by stacking up all the row vectors $A_{i\to i,j}(k)$ according to S. This leads to the following main result on the convergence of $\tilde{x}(k)$, which in turn guarantees the convergence of $\hat{x}_i(k)$ due to the convergence of $Q_{i\to i,j}(k)$.

Lemma 4. *Under Assumption 1, the estimate $\tilde{x}(k)$ converges asymptotically to $(I - A(\infty))^{-1} b(\infty)$ if the matrix $A(\infty)$ is stable (i.e., all of its eigenvalues are strictly within the unit circle). Conversely, if $A(\infty)$ is not stable, then for almost all measurements of z_i and $z_{i,j}$, $\tilde{x}(k)$ will diverge as $k \to \infty$.*

Proof. Given in Lemma 4 of the full version [23]. □

We have the following key property for $A(\infty)$.

Lemma 5. *Under Assumption 1, the diagonal elements of $A(\infty)$ are zero. Moreover, for every $(i \to i,j)$, we have*

$$A_{i\to i,j}(\infty) A_{i\to i,j}^T(\infty) \leq \rho I.$$

Proof. Given in Lemma 5 of the full version [23]. □

For a given canonical graph $\mathcal{G}$, denote by $\bar{\mathcal{G}}$ the reduced graph obtained by repeatedly removing the leaf nodes until there are no more leaf nodes (i.e., all the variable nodes are on a cycle), or $\bar{\mathcal{G}}$ is a singleton (i.e., it contains a single variable node). Without loss of generality, let the remaining nodes be $1, 2, \ldots, \bar{I}$. Further denote by $\bar{A}(\infty)$ the matrix obtained by removing the rows and columns of $A(\infty)$ associated with indices $(i \to i,j)$ for $i > \bar{I}$. For the case $\bar{\mathcal{G}}$ is a singleton, $\bar{A}(\infty)$ is void. We have the following important result:

Lemma 6. *The matrix $A(\infty)$ is stable if and only if $\bar{A}(\infty)$ is stable. In particular, $A(\infty)$ is always stable if $\mathcal{G}$ is acyclic.*

Proof. Given in Lemma 6 of the full version [23]. □

Using the result above, we can restate Lemma 4 as follows:

Theorem 2. *Under Assumption 1, every estimate $\hat{x}_i(k)$, $i = 1, 2, \ldots, I$, converges asymptotically if the matrix $\bar{A}(\infty)$ is stable. Conversely, if $\bar{A}(\infty)$ is not stable, then for almost all measurements of z_i and $z_{i,j}$, $\hat{x}_i(k)$ will diverge as $k \to \infty$.*

While the result above provides a necessary and sufficient condition for the convergence of the estimates, checking $\bar{A}(\infty)$ is not an easy task for a large system. Our next aim is to provide a sufficient condition for guaranteeing the stability of $\bar{A}(\infty)$ that is easily verifiable in a distributed fashion.

With some abuse of notation, we still denote by S a sequence of $(i \to i,j)$ for $\bar{\mathcal{G}}$. In particular, we will choose $S = \{S_1, S_2, \ldots, S_{\bar{I}}\}$ with S_i denoting a sub-sequence containing all $(i \to i,j)$ for $j \in \bar{\mathcal{N}}_i$ and $\bar{\mathcal{N}}_i$ denoting the set of neighbouring nodes of i in $\bar{\mathcal{G}}$. We further denote by $\bar{A}_i(\infty)$ the square sub-matrix of $\bar{A}(\infty)$ by keeping only the rows with indices $(i \to i,j), j \in \bar{\mathcal{N}}_i$, and only columns with indices $(j \to j,i), j \in \bar{\mathcal{N}}_i$. For better understanding of the notation, an example is given below.

Example 1. *Fig. 2 shows a simple canonical graph and the structure of the associated $A(\infty)$, where $*$ stands for a non-zero term. It is easy to verify that*

$$\begin{aligned}A_1(\infty) = A_3(\infty) &= \begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & 0 \end{bmatrix}; \\ A_2(\infty) = A_4(\infty) &= \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix}.\end{aligned}$$

We have the next result on the convergence of the estimates, which can be checked in a distributed fashion.

Theorem 3. *Recall that ρ is defined in*(22)*. Suppose there exists $\rho \leq \bar{\rho} < 1$ such that $\bar{A}_i(\infty)\bar{A}_i^T(\infty) \leq \bar{\rho} I$ for every $1 \leq i \leq \bar{I}$ which has at least three neighbouring nodes in $\bar{\mathcal{G}}$. Under Assumption 1, for every $i = 1, 2, \ldots, I$, $\hat{x}_i(k)$ converges exponentially with the rate $\bar{\rho}$. In particular, if $\mathcal{G}$ only has a single cycle (which means that every node i in $\bar{\mathcal{G}}$ has*

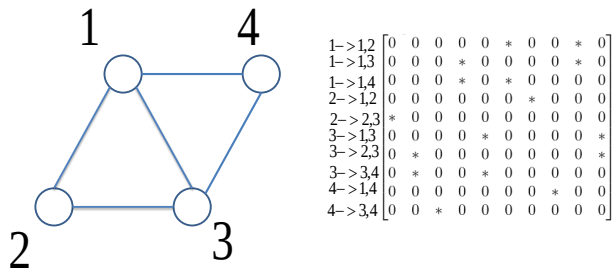


Fig. 2. An example to show the structure of $A(\infty)$

only two neighbouring nodes), $\hat{x}_i(k)$ converges exponentially with the rate of ρ.

Proof. Given in Theorem 3 of the full version [23]. □

Remark 4. Using Theorem 3, checking the convergence of the estimates amounts to computing $\bar{A}_i(\infty)$ and its maximum singular value σ_i for each node with at least three neighbouring nodes. The required $\bar{\rho}$ can be made to be $\bar{\rho} = \max_i \sigma_i^2$ using the fact that $\bar{A}_i(\infty)\bar{A}_i^T(\infty) \leq \sigma_i^2 I$. By Theorem 3, the convergence of the estimates is guaranteed if $\bar{\rho} < 1$.

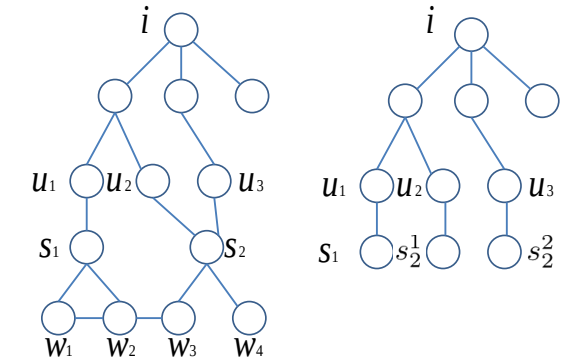
Remark 5. Algorithm 1 is designed for the static estimation problem. As for the centralized (traditional) state estimation case, this is a crucial step towards dynamic state estimation. Generalization to distributed dynamic state estimation will be a future topic. Since we have proved that the estimate using our algorithm exponentially converges, the algorithm could achieve good approximation for the optimal state estimate at time k in a few steps during the time update from k to $k+1$ and the prediction for time $k+1$ follows from neighbours' state estimates. In particular, we note that, although linear measurements (6)-(7) are assumed in this paper, it is shown in [4] that (relative) distance measurements between adjacent agents, which are seemingly nonlinear, can be effectively transformed into linear measurements when applied in distributed dynamic state estimation. Thus, our algorithm can be used in the update part of the distributed dynamic state estimation problem involving distance measurements.

VI. Accuracy Analysis on Information Matrices

In Section IV, we studied the convergence of the information matrices. In this section we study its accuracy, i.e., the difference between the information matrices $Q_i(k)$ generated by Algorithm 1 and the information matrices for the maximum likelihood estimates.

Let $\mathcal{G}_i(d)$ denote the subgraph of $\mathcal{G}$ formed by nodes which are within d hops away from node i. Denote by d_i the largest integer such that $\mathcal{G}_i(d_i)$ is acyclic. We refer to d_i as the *cycle-free depth* of node i, and $d_{\min} = \min_i d_i$ as the *cycle-free depth* of $\mathcal{G}$. If $\mathcal{G}$ is acyclic, we use the convention that $d_i = \infty$ for all i, and in this case, $\mathcal{G}_i(d_i) = \mathcal{G}$. It is emphasized that the *cycle-free depth* is a property of each vertex, and that it is upper bounded by the diameter of the graph and lower bounded by the length of the shortest cycle in the graph.

Recall that, for the measurements (6) and (7), the corresponding joint likelihood function $f(X)$ is given by (8), (9) and (10). The marginal $g_i(x_i)$ of X is given by (11). Denote by $\hat{x}_i^{ML}$ and Σ_i^{ML} the mean and covariance of x_i corresponding to $g_i(x_i)$, respectively. The superscript "ML" stands for maximum likelihood, due to the fact that $g_i(x_i)$ is Gaussian and thus $\hat{x}_i^{ML}$ is the maximum likelihood estimate of x_i. Also define the ML information matrix $Q_i^{ML} = (\Sigma_i^{ML})^{-1}$.



(a): Original graph $\mathcal{G}$ (b): Reduced graph $\tilde{\mathcal{G}}_i$

Fig. 3. Conversion of a cyclic graph into an acyclic graph

Now, for any node $i \in \mathcal{G}$, we introduce a reduced graph $\tilde{\mathcal{G}}_i$, which will hold the key to the accuracy analysis. Draw the graph $\mathcal{G}$ as depicted in Fig. 3(a). Let d_i be the cycle-free depth for node i, as defined earlier. In Fig. 3(a), $d_i = 2$. Denote by $u_1, u_2, \ldots,$ the leaf nodes of $\mathcal{G}_i(d_i)$ and denote by $s_1, s_2, \ldots,$ the nodes outside of $\mathcal{G}_i(d_i)$ which are connected to the leaf nodes and called *child* nodes. There are two cases for each child node: It connects to either one leaf node only or multiple leaf nodes. In Fig. 3(a), s_1 connects to one leaf node of $\mathcal{G}_i(d_i)$(i.e., u_1) and s_2 connects to two leaf nodes of $\mathcal{G}_i(d_i)$(i.e., u_2 and u_3). The reduced graph $\tilde{\mathcal{G}}_i$ is depicted in Fig. 3(b). This is done as follows: For each child node s, firstly remove all of its connecting nodes that are not in $\mathcal{G}_i(d_i+1)$. Then, if s is connected to multiple leaf nodes of $\mathcal{G}_i(d_i)$, split s into multiple copies, one for each connecting leaf node. In Fig. 3(b), $w_1, \ldots, w_4$ are all removed and s_2 is split into s_2^1 and s_2^2. For each child node s_i that connects to only one leaf node of $\mathcal{G}_i(d_i)$, let its self measurement in the reduced graph $\tilde{\mathcal{G}}_i$ be $z_{s_i} = C_{s_i} x_{s_i} + v_{s_i}$ with noise covariance R_{s_i} for v_{s_i}. Then, for each child node s_i^t that connects to p leaf nodes of $\mathcal{G}_i(d_i)$, we take

$$z_{s_i^t} = C_{s_i} x_{s_i^t} + v_{s_i^t} = z_{s_i}$$

with noise covariance pR_{s_i} for $v_{s_i^t}$, where p is the number of connecting leaf nodes of $\mathcal{G}_i(d_i)$ for s_i.

We have the following result, which shows that the information matrices from Algorithm 1 converge to ML information matrices exponentially as the cycle-free depths increase.

Theorem 4. *Under Assumption 1, for every node i in $\mathcal{G}$, we have*

$$0 \leq Q_i(d_i) - Q_i^{ML} \leq \tilde{\alpha}_i \tilde{\rho}_i^{d_i - 1} Q_i^{ML}, \tag{28}$$

where $\tilde{\alpha}_i$ and $\tilde{\rho}_i$ are similar to α in (23) and ρ in (22), but for the reduced graph $\tilde{\mathcal{G}}_i$.

Proof. Given in Theorem 4 of the full version [23]. □

Using Theorem 1, the result in Theorem 4 can be stated in a different way in terms of the accuracy of $Q_i(\infty)$.

Corollary 1. *Recall that $\tilde{\alpha}_i, \tilde{\rho}_i, \alpha, \rho$ are defined in the Theorem 4 and d_i is the cycle-free depth of node i. Under Assumption 1, we have, for every node i in $\mathcal{G}$,*

$$-\frac{\alpha\rho^{d_i-1}}{1+\alpha\rho^{d_i-1}}Q_i^{ML} \le Q_i(\infty)-Q_i^{ML} \le \tilde{\alpha}_i\tilde{\rho}_i^{d_i-1}Q_i^{ML}. \quad (29)$$

In particular, as $d_i \to \infty$, $Q_i(\infty) \to Q_i^{ML}$.

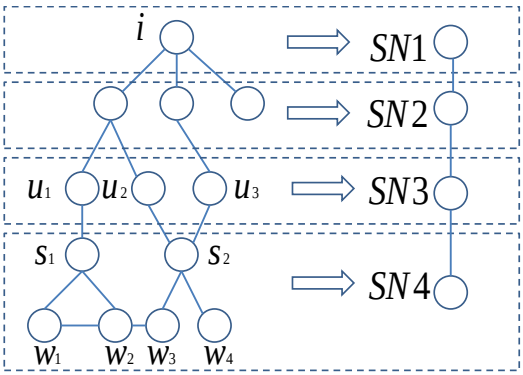
Proof. Given in Corrolary 1 of the full version [23]. □

Remark 6. We mention two properties about the $\tilde{\alpha}_i$ and $\tilde{\rho}_i$. Firstly, it is clear that for an acyclic graph $\mathcal{G}$, $\tilde{\alpha}_i = \alpha$ and $\tilde{\rho}_i = \rho$ for any i. Secondly, since each reduced graph is for a given node i and is typically a small graph (with cycle-free depth of $d_i + 1$), $\tilde{\alpha}_i$ and $\tilde{\rho}_i$ can be computed quickly (with $d_i + 1$ iterations).

VII. Accuracy Analysis for the Estimates

In this section, we study the accuracy of the state estimate from Algorithm 1. Our goal is to characterize explicitly how the distributed state estimate accuracy for node i is related to its cycle-free depth d_i.

Without loss of generality and for notational simplicity, we study node 1 in this section. Let d_1 be the cycle-free depth of node 1. We can redraw the original graph $\mathcal{G}$ (i.e., Fig. 3(a)) as a d_1+2 layer graph in Fig. 4(a), in which node 1 is placed on the top layer (layer 1), followed by all the nodes one lop away from node 1 as layer 2, then by all the nodes two hops away from node 1 as layer 3, and so on, until layer (d_1+1) which contains all the nodes d_1 hops away from node 1. All other nodes (i.e., nodes outside of $\mathcal{G}_1(d_1)$) are lumped into layer d_1+2. This graph can then be redrawn again as a line graph $\mathcal{G}^l$(i.e., Fig. 4(b)) by grouping all the nodes in layer i as a super node i (denoted by SN i) with state $\check{x}_i$ which is formed by stacking up all the states in layer i. In particular, $\check{x}_1 = x_1$. The self measurement for super node i will be denoted by $\check{z}_i$, $i = 1, 2, \ldots, d_i+2$, and the edge measurement between super nodes i and $i+1$ will be denoted by $\check{z}_{i,i+1}$, $i = 1, 2, \ldots, d_i+1$. The notations of $\check{C}_i, \check{C}_{i,j}, \check{R}_i$ and $\check{R}_{i,j}$ are similarly defined. For notational simplicity, we denote $d_1 + 2$ by n.



(a): Original graph $\mathcal{G}$ (b): Line graph $\mathcal{G}^l$

Fig. 4. Conversion of a graph with $d_1 = 2$ into a 4-layer line graph

Denoting $\check{x} = \text{col}\{\check{x}_1, \check{x}_2, \ldots, \check{x}_n\}$, we have the following result on the maximum likelihood estimate for $\check{x}$ in the trimmed line graph $\mathcal{G}^l$(i.e., Fig. 4(b)).

Lemma 7. *The maximum likelihood estimate $\check{x}^{ML}$ for $\check{x}$ is given by the solution of*

$$A\check{x}^{ML} = B, \quad (30)$$

where $A > 0$ is a tri-diagonal block matrix defined by

$$\begin{aligned}
A_{11} &= \check{C}_1^T\check{R}_1^{-1}\check{C}_1 + \check{C}_{1,2}^T\check{R}_{1,2}^{-1}\check{C}_{1,2}\\
A_{ii} &= \check{C}_i^T\check{R}_i^{-1}\check{C}_i + \check{C}_{i,i+1}^T\check{R}_{i,i+1}^{-1}\check{C}_{i,i+1}\\
&\quad + \check{C}_{i,i-1}^T\check{R}_{i-1,i}^{-1}\check{C}_{i,i-1}, \quad i = 2, \ldots, n-1\\
A_{nn} &= \check{C}_n^T\check{R}_n^{-1}\check{C}_n + \check{C}_{n,n-1}^T\check{R}_{n-1,n}^{-1}\check{C}_{n,n-1}\\
A_{i(i+1)} &= \check{C}_{i,i+1}^T\check{R}_{i,i+1}^{-1}\check{C}_{i+1,i}, \quad i = 1, 2, \ldots, n-1\\
A_{(i+1)i} &= A_{i(i+1)}^T, A_{ij} = 0, \quad |i-j| > 1,
\end{aligned}$$

and $B = \text{col}\{B_1, B_2, \ldots, B_n\}$ with

$$\begin{aligned}
B_1 &= \check{C}_1^T\check{R}_1^{-1}\check{z}_1 + \check{C}_{1,2}^T\check{R}_{1,2}^{-1}\check{z}_{1,2}\\
B_i &= \check{C}_i^T\check{R}_i^{-1}\check{z}_i + \check{C}_{i,i-1}^T\check{R}_{i-1,i}^{-1}\check{z}_{i-1,i}\\
&\quad + \check{C}_{i,i+1}^T\check{R}_{i,i+1}^{-1}\check{z}_{i,i+1}, \quad i = 2, \ldots, n-1\\
B_n &= \check{C}_n^T\check{R}_n^{-1}\check{z}_n + \check{C}_{n,n-1}^T\check{R}_{n-1,n}^{-1}\check{z}_{n,n-1}.
\end{aligned}$$

Proof. Given in Lemma 7 of the full version [23]. □

Next we give an alternative characterization for the state estimate $\hat{x}_1(d_1)$.

Lemma 8. *Under Assumption 1, the state estimate $\hat{x}_1(d_1)$ from Algorithm 1 is given by the first block of $\check{\mathbf{x}}$ which solves*

$$\mathbf{A}\check{\mathbf{x}} = \mathbf{B}, \quad (31)$$

where $\mathbf{A}$ is obtained from A by removing its last row block and last column block, and $\mathbf{B}$ is obtained from B by removing its last row block.

Proof. Given in Lemma 8 of the full version [23]. □

The next result characterizes the estimation error of the state estimate $\hat{x}_1(d_1)$.

Lemma 9. *Let $\Delta x_1(d_1) = \hat{x}_1(d_1) - x_1^{ML}$ be the estimation error for node 1. Then, under Assumption 1, we have*

$$\begin{aligned}
&\Delta x_1(d_1)\\
=&(-\tilde{A}_{11}^{-1}A_{12})\ldots(-\tilde{A}_{(n-1)(n-1)}^{-1}A_{(n-1)n})\check{x}_n^{ML}, \quad (32)
\end{aligned}$$

where

$$\begin{aligned}
\tilde{A}_{11} &= A_{11},\\
\tilde{A}_{ii} &= A_{ii} - A_{(i-1)i}^T\tilde{A}_{(i-1)(i-1)}^{-1}A_{(i-1)i}
\end{aligned}$$

for all $i = 2, \ldots, n-1$.

Proof. Given in Lemma 9 of the full version [23]. □

Theorem 5. *Under Assumption 1, for node 1 in the graph $\mathcal{G}$ with cycle-free depth d_1, we have*

$$\Delta x_1(d_1)^T Q_1(1)\Delta x_1(d_1) \le \kappa\eta^{d_1}, \quad (33)$$

with $\eta < 1$ *defined in* (14) *and*

$$\kappa = \sum_{(t,j)} (x_j^{ML})^T C_{j,t}^T R_{t,j}^{-1} C_{j,t} x_j^{ML}$$

where (t, j) *are such that node* t *is* d_1 *hops away from node* 1 *and node* j *is connected to node* t *but* $d_1 + 1$ *hops away from node* 1.

Proof. Given in Theorem 5 of the full version [23]. □

Similar to that in Corollary 1, the accuracy of $\Delta x_1(\infty)$ is given in the Corollary 2.

Corollary 2. *Let*[2]

$$B(k) = Q^{1/2}(k+1)A(k)Q^{-1/2}(k)$$

with $A(k)$ *defined in* (27) *and*

$$Q(k) = \text{diag}\left(Q_{i\to i,j}(k) : (i \to i, j) \in S\right).$$

Denote the maximum eigenvalue of $B(\infty)$ *by* $\beta = \overline{\text{eig}}\left(B(\infty)\right) < 1$. *Under Assumption 1, if* d_1 *is large enough so that* $\rho^{d_1-1} \simeq 0$ *and* $\beta^{d_1-1} \simeq 0$, *then*

$$\left\|\Delta x_1(\infty)\right\|^2 \lesssim \kappa\eta^{d_1} \left\|Q_1^{-1}(1)\right\|.$$

Proof. Given in Corollary 2 of the full version [23]. □

Remark 7. The results in Theorem 5 and Corollary 2 show in a very quantitative way that the accuracy of the state estimate depends explicitly on 1) The number of links connecting $\mathcal{G}_i(d_i)$ and outside; 2) the "size" of the state for each such node as measured by $(x_j^{ML})^T C_{j,t}^T R_{t,j}^{-1} C_{j,t} x_j^{ML} \approx x_j^T C_{j,t}^T R_{t,j}^{-1} C_{j,t} x_j$; 3) the decay rate η; 4) cycle-free depth d_i. Accurate state estimates require a combination of fast decay rate, large cycle-free depth, small number of links connecting the inside and outside of the cycle-free region, and small state "sizes" for such nodes.

VIII. Conclusion

In this paper, a BP-based distributed static estimator for large-scale networked systems is expressed in Algorithm 1. By viewing its iterations as a dynamic process, we have carried out a complete analysis for its convergence and accuracy. We have given conditions under which the Algorithm 1 is guaranteed to converge, and we have provided concrete characterizations of its accuracy. The influence of the so-called *cycle-free depth* of each node to the accuracy is exploited. As explained in the Remark 5, the Algorithm 1 can also be effective in the dynamic state estimation by running several iterations during the time update of system sampling. Our results are expected to yield a theoretical understanding of the distributed state estimation and may generate more applications for this powerful algorithm.

[2]recall from Section V that S is an ordered sequence of all $(i \to i, j)$

References

[1] U. Khan, S. Kar, and J. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.

[2] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 50–61.

[3] C. Ravazzi, N. P. Chan, and P. Frasca, "Distributed estimation from relative measurements of heterogeneous and uncertain quality," *IEEE transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 203–217, 2018.

[4] S. Battilotti, F. Cacace, M. D'Angelo, and A. Germani, "Cooperative filtering with absolute and relative measurements," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7182–7187.

[5] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[6] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[7] D. Marelli and M. Fu, "Distributed weighted least-squares estimation with fast convergence for large-scale systems," *Automatica*, vol. 51, pp. 27–39, 2015.

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

[9] J. Pearl, *Probabilistic reasoning in intelligent systems*. Morgan Kaufman, 1988.

[10] M. Fu, "Stochastic analysis of turbo decoding," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 81–100, 2005.

[11] S. Tatikonda and M. I. Jordan, "Loopy belief propogation and gibbs measures," *arXiv preprint arXiv:1301.0605*, 2012.

[12] S. Tatikonda, "Convergence of the sum-product algorithm," in *Proceedings of IEEE Information Theory Workshop*, 2003, pp. 222–225.

[13] A. Ihler, J. Fisher, and A. Willsky, "Message errors in belief propagation," in *Advances in Neural Information Processing Systems*, 2004, pp. 609–616.

[14] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4422–4437, 2007.

[15] A. T. Ihler, W. F. John III, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *Journal of Machine Learning Research*, vol. 6, pp. 905–936, 2005.

[16] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *The Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.

[17] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[18] Q. Su and Y.-C. Wu, "Convergence analysis of the variance in Gaussian belief propagation," *IEEE Transactions on Signal Processing*, vol. 62, pp. 5119–5131, 2014.

[19] ——, "On convergence conditions of Gaussian belief propagation," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1144–1155, 2015.

[20] D. Malioutov, J. Johnson, and A. Willsky, "Walk-sums and belief propagation in gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.

[21] C. Moallemi and R. Benjamin, "Convergence of min-sum message-passing for convex optimization," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 2041–2050, 2010.

[22] J. Du, S. Ma, Y.-C. Wu, S. Kar, and J. M. Moura, "Convergence analysis of distributed inference with vector-valued gaussian belief propagation," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6302–6339, 2017.

[23] T. Sui, M. Fu, D. Marelli, and X. Sun, "Convergence and accuracy analysis for a distributed static state estimator based on gaussian belief propagation," *arXiv:1806.09104*, 2020.

[24] X. Tai, Z. Lin, M. Fu, and Y. Sun, "A new distributed state estimation technique for power networks," in *American Control Conference*, 2013, pp. 3338–3343.

[25] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural computation*, vol. 12, pp. 1–41, 2000.