

Robust Distributed Estimation Using the Embedded Subgraphs Algorithm

Véronique Delouille, *Member, IEEE*, Ramesh “Neelsh” Neelamani, *Member, IEEE*, and Richard G. Baraniuk, *Fellow, IEEE*

Abstract—We propose a new iterative, distributed approach for linear minimum mean-square-error (LMMSE) estimation in graphical models with cycles. The *embedded subgraphs algorithm* (ESA) decomposes a loopy graphical model into a number of linked embedded subgraphs and applies the classical parallel block Jacobi iteration comprising local LMMSE estimation in each subgraph (involving inversion of a small matrix) followed by an information exchange between neighboring nodes and subgraphs. Our primary application is sensor networks, where the model encodes the correlation structure of the sensor measurements, which are assumed to be Gaussian. The resulting LMMSE estimation problem involves a large matrix inverse, which must be solved in-network with distributed computation and minimal intersensor communication. By invoking the theory of asynchronous iterations, we prove that ESA is robust to temporary communication faults such as failing links and sleeping nodes, and enjoys guaranteed convergence under relatively mild conditions. Simulation studies demonstrate that ESA compares favorably with other recently proposed algorithms for distributed estimation. Simulations also indicate that energy consumption for iterative estimation increases substantially as more links fail or nodes sleep. Thus, somewhat surprisingly, sensor network energy conservation strategies such as low-powered transmission and aggressive sleep schedules could actually prove counterproductive. Our results can be replicated using MATLAB code from www.dsp.rice.edu/software.

Index Terms—Asynchronous iterations, distributed estimation, graphical models, matrix splitting, sensor networks, Wiener filter.

I. INTRODUCTION

SENSORS, signal processing algorithms, and wireless communication technologies have matured to the point where large networks of sensor nodes can now be easily deployed in a wide variety of environments, making them very attractive for large-scale applications like environmental monitoring, security surveillance, and disaster relief [1]. Often battery-powered sensor nodes are capable of *sensing*, *computing*, and *communicating* information.

Manuscript received January 20, 2005; revised July 29, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Venkatesh Saligrama. This work was supported by grants from NSF, AFOSR, ONR, DARPA, and the Texas Instruments Leadership University Program.

V. Delouille is with the Royal Observatory of Belgium, B-1180 Bruxelles, Belgium (e-mail: veronique.delouille@oma.be).

R. Neelamani is with the ExxonMobil Upstream Research Company, Houston, TX 77027 USA (e-mail: ramesh.neelamani@exxonmobil.com).

R. G. Baraniuk is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: richb@rice.edu; Website: www.dsp.rice.edu).

Digital Object Identifier 10.1109/TSP.2006.874839

In the setting considered here, we assume that N sensors make noisy scalar measurements of their respective physical environments,¹ such as the local temperature, wind speed, or concentration of some substance. We model the vector of N measurements y as a Gaussian random field $y = x + \epsilon$, with

$$y_i = x_i + \epsilon_i, \quad i = 1, 2, \dots, N. \quad (1)$$

Here, x_i is the true value of the random field at sensor i , and ϵ_i is sensor i 's measurement noise. We assume that the vector x is jointly Gaussian with zero mean and known covariance matrix Σ and that the measurement noise ϵ is independent of x and Gaussian with zero mean and diagonal covariance matrix R . A core sensor network problem involves producing accurate estimates of the true x from the noisy y .

Under the framework (1), a natural estimate of x is the linear minimum mean-square-error (LMMSE) Wiener estimate \hat{x} obtained by solving the *normal equations*²

$$V\hat{x} = R^{-1}y \quad (2)$$

$$\text{with } V := \Sigma^{-1} + R^{-1}. \quad (3)$$

In a sensor network, a naive approach to obtain \hat{x} would first transmit all of the measurements y to a central location, and then compute \hat{x} by applying the $N \times N$ matrix inverse V^{-1} to $R^{-1}y$. Unfortunately, since communication typically consumes significant energy relative to computation energy [2], this centralized approach can be extremely energy intensive and thereby infeasible for large sensor networks.

In this paper, we develop an algorithm that computes the LMMSE estimate \hat{x} by diffusing information across the network, effectively distributing the $N \times N$ matrix inversion across all sensors. This algorithm is parallel, involves communication only among neighboring sensors, and expends minimal communication energy. Moreover, it computes the LMMSE estimate “in place”; that is, eventually each sensor has a good estimate of the environment it is sensing.

Graphical models (GMs) provide a natural framework both for modeling the correlations amongst the sensor measurements and for developing efficient distributed estimation algorithms [3]. In a sensor network GM, the nodes correspond to the variables x_i , and the existence of an edge between nodes x_i and x_j

¹All discussions easily extend to the case of vector sensor measurements.

²The techniques described here can be easily extended to obtain the LMMSE estimate for non-Gaussian data and noise as well because the expression for the LMMSE estimate (2), (3) remains the same. However, for non-Gaussian data, the LMMSE estimate is not necessarily the minimum mean-square-error estimate.

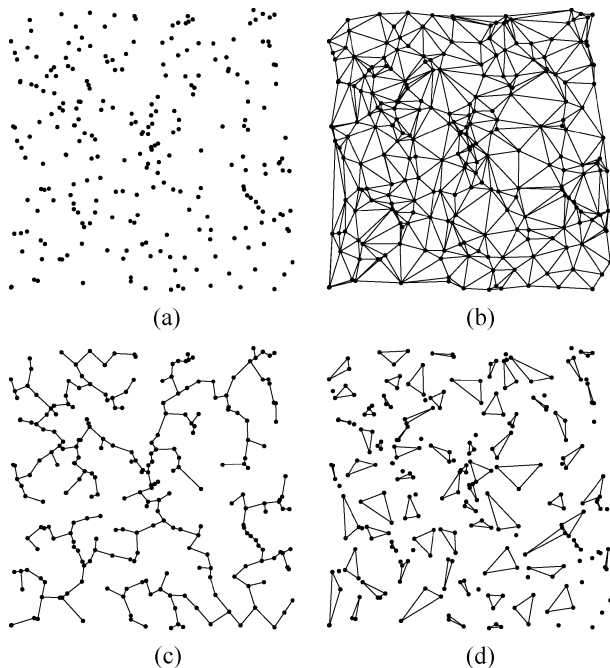


Fig. 1. (a) Example sensor network with 250 uniformly distributed sensor nodes. (b) Graphical model for the hidden variables x_i based on the Delaunay triangulation of the sensor locations. (c) Minimum spanning tree embedded in the Delaunay graph. (d) A set of disjoint subgraphs. Each subgraph is a triangle or a singleton node embedded in the Delaunay triangulation. Disjoint refers to the fact that no two subgraphs share a common node.

indicates a conditional dependency or nonzero *partial correlation*. For a Gaussian GM, this means that the (i, j) th element of Σ^{-1} is nonzero [3]. In sensor network applications, the quantities of interest (for example, temperature or gas concentration distributions) are often locally smooth. In such cases, the true field's correlation could be modeled using a GM where each node is connected only to the nodes that are spatially close (say, with a certain geographic radius). For example, Fig. 1(b) illustrates a GM based on a *Delaunay triangulation* [4] of the node locations from Fig. 1(a).³

There exist simple and efficient distributed LMMSE estimation algorithms such as generalizations of the Kalman filter for GMs without loops (for example, Markov chains and tree-structured graphs) [7], [8]. Unfortunately, the situation is anything but straightforward when the GM contains loops (for example, see Fig. 1(b)). Proposed distributed estimation approaches for loopy GMs include noniterative Gaussian elimination message passing [9] as well as iterative methods like belief propagation (BP) [10], loopy BP [11], and embedded trees [7], [12]. Since only spatially close nodes are connected in typical sensor network GMs, the above estimation algorithms require only local communication (that is, communication between nearby nodes only). However, each of these algorithms either lacks fault tolerance, lacks parallelism, or converges slowly, which limits their applicability to wireless sensor networks.

Our proposed method is to employ the classical block Jacobi algorithm [13], [14] to solve the normal equations (2), (3). We

³A Delaunay triangulation has the attractive property of linking together the closest nodes. Note that it can be computed in a distributed way [5], [6].

term the resulting approach *embedded subgraphs algorithm* (ESA). Given a loopy GM, ESA first decomposes it into a number of linked *embedded subgraphs* [see Fig. 1(d) for an example of *embedded subgraphs* where each subgraph is either a triangle or a singleton node]. Then, ESA computes the solution to (2) via a two-step iteration based on *matrix splitting* [13], [14].

Update: The nodes in each subgraph update the nodes in the neighboring subgraphs with their current estimate of x .

Solve: Within each subgraph, the nodes communicate amongst themselves to solve for a new estimate of x (for example, each embedded triangle performs a 3×3 matrix inverse computation).

ESA is well suited for wireless sensor networks, because it is local and parallel: only close-by sensors exchange information, and each subgraph performs the solve step independently. Further, ESA is scalable and energy efficient.

We prove below that ESA iterations converge to the LMMSE estimate \hat{x} under mild conditions. We also prove that convergence holds when the noise variance is small or when the x 's partial correlations are nonnegative. We also develop a simple modification of ESA—termed *Robust ESA*—that is resilient to communication failures. One of our key theoretical contributions is the analysis of Robust ESA's convergence using existing results on asynchronous iterations [15]. We identify general conditions under which Robust ESA converges to the LMMSE estimate. Further, similar to the fault-free case, we show that convergence holds when the noise variance is small or when the x 's partial correlations are nonnegative.

Our simulations substantiate that ESA compares favorably to previously proposed techniques such as embedded trees [7], [12] and loopy BP [11]. Our results can be replicated using MATLAB code from www.dsp.rice.edu/software. Further, our experiments indicate that networks with nodes that sleep (shut down periodically ostensibly to conserve energy) can actually consume *more* total energy to converge to the LMMSE estimate due to the required additional iterations. To save energy, it is critical for the nodes to synchronize so that each node is aware of whether its neighbors are awake or asleep.

This paper is organized as follows. After defining GMs and our estimation problem more precisely in Section II, we review matrix splitting algorithms in Section III and develop ESA in Section IV. Section V compares the performance of ESA against loopy belief propagation, embedded trees, and Jacobi for the fault-free case. Section VI overviews Robust ESA's fault tolerance and theoretical convergence properties, while Section VII demonstrates the robustness of ESA to faults such as sleeping nodes and failing communication links. We conclude with a discussion and suggestions for future work in Section VIII.

II. GRAPHICAL MODELS AND SENSOR NETWORKS

GMs provide a convenient foundation to develop distributed LMMSE estimation algorithms for sensor networks. The terms “sensor” and “node” are used interchangeably below.

A. Graphical Models

GMs, including Bayesian networks and Markov random fields (MRFs), represent statistical dependencies between a set of variables by means of a graph [3]. Let \mathcal{G} be an undirected graph defined by a set of nodes \mathcal{V} and a set of edges \mathcal{E} (see Fig. 1). Nodes i and j are termed *neighbors* if they are connected by an edge; that is, $(i, j) \in \mathcal{E}$. A loop consists of a set of nodes i_1, i_2, \dots, i_n such that nodes i_k is i_{k+1} 's neighbor, $k = 1, 2, \dots, n-1$, and i_1 is i_n 's neighbor.

Each node $i \in \mathcal{V}$ is associated with an unobserved or *hidden variable* x_i as well as with a noisy *measurement* y_i . Since we have assumed that the noise ϵ in (1) is independent of x , given the set of hidden variables $x := \{x_i | i \in \mathcal{V}\}$, the observations $y := \{y_i | i \in \mathcal{V}\}$ are independent of each other. A missing edge (i, j) between nodes i and j implies conditional independence between the variables x_i and x_j given all other hidden variables. Let $p(\cdot)$ denote probabilities and $\mathcal{V} \setminus i$ denote the set obtained by excluding i from \mathcal{V} . The *neighborhood* of node i is defined as $\mathcal{N}(i) := \{j | (i, j) \in \mathcal{E}\}$. Then, $p(x_i | x_{\mathcal{V} \setminus i}) = p(x_i | x_{\mathcal{N}(i)})$.

In this paper, we focus on the case where the target variables x form a jointly Gaussian process to be estimated. In this case, the entries in the so-called *precision matrix* $Q := \Sigma^{-1}$ are related to the *partial correlation* between x_i and x_j by $\text{Corr}(x_i, x_j | x_{\mathcal{V} \setminus \{i, j\}}) = -Q_{ij} / \sqrt{Q_{ii}Q_{jj}}$ [3].⁴ A Gaussian GM contains no edge between x_i and x_j if and only if $Q_{i,j} = 0$. We say that a GM is *locally connected* when each variable x_i is uncorrelated with the rest of the GM given the neighboring measurements x_j . Locally connected GMs continue to indirectly capture correlations between distant nodes: even though the underlying Q is sparse (many zero entries), the covariance matrix $\Sigma = Q^{-1}$ will generally be full (no zero entries). In particular, if x_i and x_j are partially correlated and so are x_j and x_k , then x_i and x_k are correlated through x_j .

B. Graphical Models for Sensor Networks

In many sensor network estimation applications, the quantities of interest, such as temperature, wind speed, or concentration of some substance, are often locally smooth. Such quantities can be effectively modeled by loopy, locally connected GMs, in which only spatially neighboring nodes are connected by edges. (By analogy, locally smooth images have been successfully modeled using Gaussian Markov random-field GMs that connect only adjacent pixels [16].)

For example, Fig. 1(b) illustrates a GM based on a *Delaunay triangulation* [4] of the node locations from Fig. 1(a).⁵ As we will see below, locally connected GMs lead directly to efficient distributed estimation algorithms.

In this paper, we assume that the GM $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the $Q = \Sigma^{-1}$ are known. In practice, the GM might need to be estimated

⁴The correlation between two univariate, zero-mean random variables x and y is $\text{Corr}(x, y) := \mathbb{E}(x, y) / \sqrt{\mathbb{E}(x^2)\mathbb{E}(y^2)}$, with \mathbb{E} denoting the expectation operator.

⁵A Delaunay triangulation links the *closest* neighbors in a graph to form triangles in the sense that the circumcircle of each triangle—the circle passing through the three vertices of the triangle—does not contain any points of the triangulation.

from the data. In sensor network estimation applications, the estimated GM should balance the tradeoff between accurately capturing the correlation structure of the quantities being measured and supporting energy-efficient distributed algorithms. Furthermore, the GM estimation algorithm itself should be distributed and communication efficient. For the restrictive case when the graph \mathcal{G} is *decomposable*,⁶ the nonzero entries of Q can be estimated in an efficient way [3, p. 146]. Although the model estimation problem is well-understood for arbitrary GMs [17], there do not exist efficient techniques to perform the model estimation.

III. ITERATIVE DISTRIBUTED ESTIMATION

We desire to exploit a sensor network's computation and communication capabilities to solve the linear system (2) for a loopy GM in a distributed, energy-efficient way. If the underlying GM is loop-free, then the sensor network can invoke existing noniterative approaches such as generalized Kalman filtering algorithms [7], [8] to solve (2). Such algorithms are energy efficient and distributed because each node needs to communicate just twice with every neighbor in the GM. However, for loopy GMs, energy-efficient distributed LMMSE estimation is poorly understood. Similar to the naive centralized approach (see Introduction), noniterative approaches such as extended message passing algorithm based on Gaussian elimination [9] are extremely communication energy intensive for loopy GMs and thereby inappropriate for sensor networks. In order to efficiently handle loopy GMs, we consider here an attractive class of iterative methods for solving (2) based on a clever decomposition of V .

A. Matrix-Splitting Algorithms

Matrix splitting is an iterative approach to solve the linear system (2) [13], [14]. Let $\bar{y} = R^{-1}y$ denote the normalized observation vector (R is diagonal). Then, (2) becomes

$$V\hat{x} = \bar{y}, \quad (4)$$

If we rewrite V by “splitting” it into

$$V = J - K \quad (5)$$

then solving (4) is equivalent to finding a fixed point of the system [13], [14]

$$J\hat{x} = \bar{y} + K\hat{x}.$$

Specializing matrix splitting to LMMSE estimation in loopy sensor network GMs, we assume that each sensor $i \in \mathcal{G}$ has an initial estimate \hat{x}_i^0 and knows row i of the matrix V . Starting

⁶A graph is decomposable if and only if all cycles of length ≥ 4 contain a chord; that is, the cycle contains two nonconsecutive nodes that are connected [3].

from the initial guess \hat{x}^0 , we solve the fixed point equation by generating a sequence of iterates $\{\hat{x}^m\}_{m=1}^{\infty}$ according to the following two-step recursion:

Update: Each node i exchanges its estimate \hat{x}_i^{m-1} with the set of neighbors $\{j \in \mathcal{N}(i) | K_{i,j} \neq 0\}$. Then, it updates an auxiliary variable \hat{y}_i^m according to

$$\hat{y}_i^m = \bar{y}_i + K \hat{x}^{m-1} \quad (6)$$

Solve: Each nodes i exchanges the updated \hat{y}_i^m with the set of neighbors $\{j \in \mathcal{N}(i) | J_{i,j} \neq 0\}$ and solves for the new estimate \hat{x}_i^m according to

$$\hat{x}^m = J^{-1} \hat{y}^m. \quad (7)$$

The iterations are terminated when $\|\hat{x}^m - \hat{x}^{m-1}\|_2 / \|\hat{x}^m\|_2$ falls below a chosen threshold.

The following theorem identifies the necessary and sufficient conditions for this iteration to converge. The *spectral radius* $\rho(J^{-1}K)$ is defined as the magnitude of the maximum eigenvalue of $J^{-1}K$.

Theorem 1 [13], [14]: Matrix-splitting iterations converge to \hat{x} for any arbitrary initial guess \hat{x}^0 if and only if the spectral radius $\rho(J^{-1}K) < 1$.

The spectral radius also affects the rate of convergence (with respect to the number of iterations), with smaller ρ leading to faster convergence [13].

For sensor networks, we seek a (J, K) pair for which the solution to this fixed-point problem can be found iteratively by simple computations and information exchanges between neighboring nodes. Sections III-B, III-C, and IV discuss appropriate choices of (J, K) for distributed estimation. For locally connected GMs, these splitting choices require information exchanges only between neighboring nodes.

B. Jacobi Algorithm

The simplest splitting algorithm is the well-known *Jacobi algorithm* that sets J equal to the diagonal of V [13], [14]

$$J = \begin{bmatrix} V_{1,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & V_{N,N} \end{bmatrix} \quad (8)$$

where $V_{k,k}$, $k = 1, 2, \dots, N$ denote the diagonal elements of V . This J is trivially invertible. Iteration m proceeds as follows.

Jacobi update: Each node i exchanges its estimate \hat{x}_i^{m-1} with all its neighbors $\{j \in \mathcal{N}(i)\}$. Then, it updates its auxiliary variable \hat{y}_i^m according to

$$\hat{y}_i^m = \bar{y}_i - \sum_{j \in \mathcal{N}(i)} V_{i,j} \hat{x}_j^{m-1}, \quad \forall i \in \mathcal{G}. \quad (9)$$

Jacobi solve: Each node i solves for the new estimate \hat{x}_i^m by simply scaling the updated \hat{y}_i^m by $V_{i,i}^{-1}$, the reciprocal of V 's corresponding diagonal entry

$$\hat{x}_i^m = V_{i,i}^{-1} \hat{y}_i^m \quad \forall i \in \mathcal{G}. \quad (10)$$

Jacobi converges slowly in general, but it has the advantage of being eminently *local*: each node i needs to know only its neighbors' values $x_{\mathcal{N}(i)}$ in order to proceed. It is also highly parallel and can proceed with minimal synchronized communication between nodes.

C. Embedded Trees Algorithm

The *embedded trees* algorithm chooses a novel matrix split with $J := J_T$ corresponding to a spanning tree \mathcal{G}_T embedded in the loopy graph \mathcal{G} [7], [12]. For example, Fig. 1(c) illustrates a minimum-spanning tree for the sensor network in Fig. 1(b). Iteration m proceeds as follows.

Tree update: Each node on the spanning tree \mathcal{G}_T transmits its current estimate \hat{x}_i^{m-1} to all neighbors except those that are its neighbors in \mathcal{G}_T . All nodes then update their auxiliary variables according to

$$\hat{y}_i^m = \bar{y}_i - \sum_{j \in \mathcal{N}(i) \setminus \mathcal{G}_T} V_{i,j} \hat{x}_j^{m-1} \quad \forall i \in \mathcal{G}. \quad (11)$$

Tree solve: The nodes solve for the new estimate

$$\hat{x}^m = J_T^{-1} \hat{y}^m. \quad (12)$$

by communicating a weighted combination of updated values to each other in a sequence dictated by a message passing algorithms such as belief propagation [7], [12].

Embedded trees typically converges faster than Jacobi, particularly for graphs with few loops; conditions for convergence are discussed further in [7]. Iteration over multiple different spanning trees is also possible and in certain cases further improves the rate of convergence [7]. For the special case when the graph contains few loops, [7] proposes a clever twist to efficiently compute the variance of \hat{x} that is, the diagonal elements of V^{-1} .

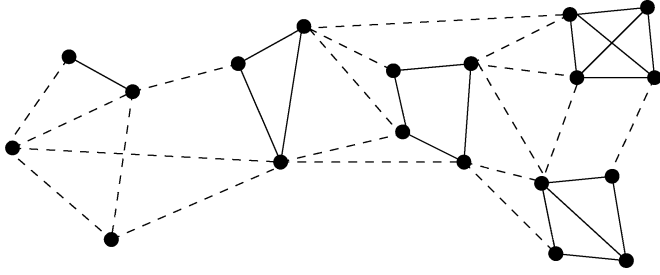


Fig. 2. Embedded subgraphs. A GM (solid and dashed lines) and its decomposition into disjoint, faithful subgraphs (solid lines). From left to right: Two singletons, a dipole, triangle, quadrangle, chordal quadrangle, and fully connected quadrangle.

In contrast to Jacobi, embedded trees is *global* requires synchronized internode communications, and lacks parallelism. In particular, during the embedded trees solve step, messages must be sequentially passed throughout the entire network before node i can compute the next value of \hat{x}_i^m .

IV. EMBEDDED SUBGRAPHS ALGORITHM

We now propose a different matrix splitting-based approach for LMMSE estimation in loopy GMs and show that it is local, parallel, demands a minimal amount of internode synchronization, and converges rapidly.

When a GM contains many loops, it becomes efficient to place in the matrix J the elements of the matrix V corresponding to small subgraphs of nodes from the graph \mathcal{G} . Consider decomposing \mathcal{G} into M disjoint subgraphs $\{s_k\}_{k=1}^M$ such that each s_k contains all edges between the s_k 's nodes that exist in \mathcal{G} . We term such a subgraph a *faithful subgraph*. The union of faithful subgraphs contains all nodes in \mathcal{V} , but no node appears in more than one subgraph. Let $s_k(i)$ denote the faithful subgraph containing node i . Examples of faithful subgraphs include singletons dipoles, triangles, quadrangles, quadrangles with chords, and so on (see Fig. 2).

For an appropriate ordering of the nodes, J becomes the block diagonal matrix

$$J = \begin{bmatrix} V_{s_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & V_{s_M} \end{bmatrix} \quad (13)$$

where V_{s_k} , $k = 1, 2, \dots, M$ denote the $L_k \times L_k$ submatrix of V corresponding to the faithful subgraph s_k of the GM. Such a splitting of J is commonly referred to as the *block Jacobi* partitioning [13], [14].

A. ESA Steps

We now introduce a sequential aspect to the block Jacobi iterations, and call the resulting algorithm ESA. Iteration m of ESA proceeds as follows.

ESA Update: Each node i exchanges its estimate \hat{x}_i^{m-1} with all of its neighbors except those belonging to its faithful sub-

graph $s_k(i)$. Then, it updates an auxiliary variable \hat{y}_i^m according to

$$\hat{y}_i^m = \bar{y}_i - \sum_{j \in \{\mathcal{N}(i) \setminus s_k(i)\}} V_{i,j} \hat{x}_j^{m-1}, \quad \forall i \in \mathcal{G}. \quad (14)$$

ESA Solve: Within the embedded subgraph $s_k(i)$, the nodes exchange their updated $\hat{y}_{s_k(i)}^m$ and solve for the new estimate $\hat{x}_{s_k(i)}^m$ according to

$$\hat{x}_{s_k(i)}^m = V_{s_k(i)}^{-1} \hat{y}_{s_k(i)}^m, \quad \forall i \in \mathcal{G}. \quad (15)$$

The inverse of each $L_k \times L_k$ matrix V_{s_k} in the ESA solve step can be computed at initialization and stored locally in each node. (If all subgraphs are singletons, then ESA reduces to the Jacobi algorithm described in Section III.B.)

All the subgraphs can run the block Jacobi iteration steps (14), (15) in parallel because they are disjoint. In contrast to the parallel block Jacobi, the Gauss–Seidel method [13], [14] is sequential: once a node has been updated, its new value is immediately used to update the next node. Consequently, Gauss–Seidel iterations typically converge faster than Jacobi [13], [14]. Hence, to improve the convergence of ESA, we slightly reduce the parallelism of block Jacobi and introduce a “Gauss–Seidel” aspect that proceeds in two stages.

In the first stage, all nodes belonging to embedded subgraphs of size $L_k \geq 2$ perform the ESA update and ESA solve steps to obtain their new estimate \hat{x}^m at iteration m . In the second stage, each singleton node i gathers the most recent estimates from its neighbors $j \in \mathcal{N}(i)$ and uses these values to update its auxiliary variable \hat{y}_i^m and solve for $\hat{x}_i^m = V_{i,i}^{-1} \hat{y}_i^m$.

B. Convergence

ESA converges to the LMMSE solution if and only if the matrix splitting is chosen such that the spectral radius $\rho(J^{-1}K) < 1$ (see Theorem 1). We now present two sufficient conditions for convergence of matrix splitting iterations. They respectively identify the signal correlation structure and the noise variance level that guarantee convergence. See Appendix I for the proofs.

Theorem 2: ESA converges to the LMMSE estimate if either of the following conditions hold:

1. the partial correlations of x in (1) are nonnegative

$$\text{Corr}(x_i, x_j | x_{\mathcal{V} \setminus \{i,j\}}) \geq 0 \quad \forall i, j \quad (16)$$

2. the variance of the noise ϵ in (1) is small

$$\mathbb{E}(\epsilon_i^2) \leq \frac{1}{2 \times \text{maximum absolute eigenvalue of } K}, \quad \forall i. \quad (17)$$

C. ESA Properties and Parameters

1) *Scalability and Self-Organization:* Since ESA operates in parallel over all disjoint subgraphs, the amount of time spent

on each iteration does not change as we increase the number of nodes N in the network. Hence, ESA is eminently scalable. Loopy BP also operates in parallel [18], and is hence scalable. However, it typically needs to transmit more messages and thereby consumes more energy.

On the other hand, since embedded trees [7] and Gaussian elimination [9] rely on sequential message passing on a tree, their time per iteration will increase with the diameter of the network,⁷ which grows with N [19].

Moreover, ESA can begin iterating without expensive upfront route finding. Indeed, we can envision simple protocols to associate the sensors into disjoint triangles; for example, refer to [20].

2) *Size of Embedded Subgraphs*: The choice of the sizes of the embedded subgraphs s_k is application -dependent. If the size $L_k > 5$, then an iterative method is needed to invert the $L_k \times L_k$ matrix used in the ESA solve step at startup. However, larger blocks V_{s_k} also increase the algorithm's rate of convergence [13, p. 199]. Hence, the tradeoff between computation/communication versus faster convergence will dictate the optimal subgraph size L_k .

Since any planar graph can be decomposed into a set of embedded triangles, dipoles, and singletons, in the simulations of Sections V and VII, we focus on triangulated GMs and study the performance of the resulting *embedded triangles algorithm* (ETA) [21].

3) *Error Variance*: The variances of the LMMSE estimates in \hat{x} are given by the diagonal elements of V^{-1} . Running ESA to solve (4) with \bar{y} replaced by the unit vector e_i , which contains a 1 in the i th position and 0's elsewhere, yields the i th row of V^{-1} , and in particular, the variance of \hat{x}_i at node i . The iteration is guaranteed to converge to the correct variance whenever the LMMSE iteration converges.

During variance estimation, the nodes need not communicate messages until they receive a nonzero message from a neighbor. In Section V-C we show that the first few iterations, which involve only a few total communications, can produce a good approximation to the variance.

V. EXPERIMENTS: FAULT-FREE CASE

This section compares the convergence speed of loopy BP, Jacobi, embedded trees, and ETA as a function of internode communication energy expenditure. We assume that all sensors are awake and can communicate with their neighbors without error; we term this the fault-free case. The results presented in this section can be replicated using MATLAB code that can be downloaded from www.dsp.rice.edu/software.

A. Data Generation

Consider 250 sensor nodes randomly distributed in the square $[0, 1] \times [0, 1]$ [see Fig. 1(a)]. We generate observations $y = x + \epsilon$, where ϵ is a zero-mean Gaussian random vector having

⁷The diameter of a graph is the length of the longest shortest path between two nodes.

a covariance matrix $\sigma^2 I$, with I the identity matrix; we chose $\sigma^2 = 4$. To generate the variables x , we set up the partial correlation coefficients $\text{Corr}(x_i, x_j | x_{\mathcal{N} \setminus \{i, j\}}) = c(1 - \text{dist}(i, j))$, with $\text{dist}(i, j)$ the Euclidean distance between nodes i and j , and c a constant for which the precision matrix $Q = \Sigma^{-1}$ remains positive definite.⁸ For the diagonal entries of Q , we use random variables uniformly distributed between 1 and 2. This permits us to compute the values $Q_{i,j}$, $i \neq j$, as described in Section II.A and to generate x as a zero-mean jointly Gaussian random vector with covariance matrix Q^{-1} . This specification of Q ensures that the elements of $\Sigma = Q^{-1}$ are positive (see Section A of Appendix I). We assume that $V = \Sigma^{-1} + R^{-1} = \Sigma^{-1} + \sigma^{-2}I$ (see (3)) is known. Our discussions below address only the error due to the iterative nature of the matrix splitting algorithm (6), (7).

B. Comparison Metrics

To evaluate the different algorithms, we compare the decay of their *residual normalized mean-squared error* (RNMSE) as a function of the total energy consumed by internode communications. We neglect the energy consumed by the computations at each node, assuming that the amount of energy required to compute one byte of information is much smaller than the amount required to communicate it to a neighbor [22]. The RNMSE at iteration m is computed as [7]

$$\text{RNMSE}^m := \|\bar{y} - V\hat{x}^m\|_2 / \|\bar{y}\|_2 \quad (18)$$

with $\bar{y} = y/\sigma^2$ and $V = \Sigma^{-1} + \sigma^{-2}I$.

In our experiments, we compute the internode communication energy assuming that the nodes are equipped with either *omnidirectional transmitters* or *unidirectional transmitters*.

With omnidirectional transmitters, a node can communicate a common value to all of its neighbors simultaneously with a single broadcast. Examples are nodes with omnidirectional radio antennas. Let $s_k(i)$ be the embedded triangle that contains node i . We assume that the communication energy λ_i^m expended by sensor i during iteration m is proportional to [2]

$$\lambda_i^m := \max_{j \in s_k(i)} (\text{dist}(i, j))^2 + \max_{j \in \{\mathcal{N}(i) \setminus s_k(i)\}} (\text{dist}(i, j))^2.$$

With unidirectional transmitters, a node must communicate with each of its neighbors one at a time, even to convey a common value. Examples are nodes with laser-based line-of-sight transmitters. In this case, we assume that the communication energy λ_i expended by sensor i during iteration m is proportional to [2]

$$\lambda_i^m := \sum_{j \in \mathcal{N}(i)} (\text{dist}(i, j))^2.$$

⁸If $c = 0$, then Q becomes diagonal and the x_i 's become uncorrelated; this is not very interesting. To ensure that the x_i 's are sufficiently correlated, we set $c = 5.74$, which was the largest value that we could find.

The total energy λ^m expended by all the sensors up to iteration m is given by

$$\lambda^m := \sum_{n=1}^m \sum_{i=1}^N \lambda_i^n.$$

C. Fault-Free Results

For the sensor network in Figs. 1(a), Fig. 3(a) and (b) displays the decay of the RNMSE with increasing energy consumption. The plots confirm experimentally that all four algorithms converge. Further, Fig. 3(a) illustrates the superior performance of ETA when the sensors are equipped with unidirectional transmitters; its RNMSE decays faster with increasing energy consumption. With omnidirectional communication, Jacobi, embedded trees and ESA perform comparably [see Fig. 3(b)]. Jacobi fully exploits omnidirectional transmitters because it simply broadcasts a common value to all its neighbors during its update step. In contrast, embedded trees and ETA expend energy communicating different messages to the neighboring nodes. (For the fault-free case, the use of multiple spanning trees in the embedded trees algorithm could yield improved performances.) Compared to ESA, BP converges slower by more than a factor of two. In contrast to ESA, Jacobi, and embedded trees, during BP iterations, each node needs to communicate *two* messages (the estimate and its variance) to the neighbors instead of *one*. This increased communication partially explains BP's slow convergence. In return, BP computes not only the LMMSE estimate, but also provide a number for its variance (this number underestimates the variance, but it can be corrected [11]). Note that loopy BP does not exploit any omnidirectional communication capabilities since a node transmits a different message to each of its neighbors (see [18, sec. 2.3.2]).

We estimated the variance of a centrally located node in Fig. 1 as described in Section IV-C-3 by transmitting only the nonzero messages. We stopped the iterations when the first "echo" reached the node from the network's boundary. With unidirectional transmitters, by spending just 5% of the energy required to estimate \hat{x} up to a precision of 10^{-3} [that is, by spending $250 \cdot 5\% = 12.5$ units; refer to Fig. 3(a)], this scheme estimated the variance with a precision of 10^{-2} .

VI. FAULT TOLERANCE

We now discuss how Jacobi and ESA can be robustified to handle temporary faults such as sleeping nodes and failing internode communication links. We prove that both of the robustified algorithms are guaranteed to converge under mild conditions.

In our opinion, it seems more difficult to make embedded trees (with single or multiple spanning trees) resilient to sleeping nodes and failing links. During the embedded trees' solve step, messages must be sequentially and successfully transmitted through the entire tree to compute the new \hat{x}^m . If a link (i, j) belonging to the tree fails, then it is unclear how to proceed with the iteration until the tree is healed (perhaps by some higher level networking protocol).

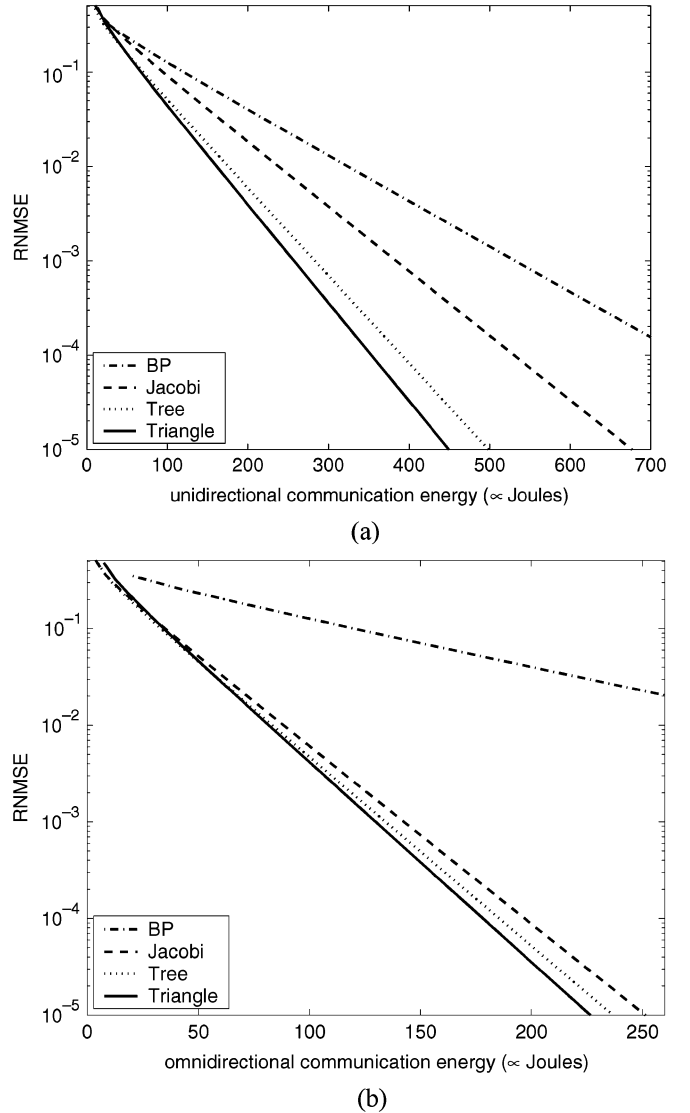


Fig. 3. Fault-free case. Simulation results for distributed estimation with the sensor network from Fig. 1(b) using loopy BP, Jacobi, embedded trees, and ETA. (a) RNMSE versus energy (unidirectional communication); (b) RNMSE versus energy (omnidirectional communication).

A. Sleeping Nodes and Failing Links

In practical sensor networks, internode communications can fail due to two main causes. A node might stop interacting temporarily with its neighbors because it powers down to conserve energy; such nodes are referred to as *sleeping nodes* [23], [24]. Sometimes a transmitted packet of information could be corrupted by temporary failure of a communications link (for example, due to channel fluctuations caused by multipath fading); such links are referred to as *failing links*. (A sleeping node can be modeled as having failing communication links to all of its neighbors.)

We consider only temporarily failing links and sleeping nodes. In the case of permanent link failures or dead sensors, the GM must be updated. Methods to handle such faults, including imputation for missing data [25, p. 200] are beyond the scope of this paper.

Sleeping nodes and failing links will clearly affect the convergence performance of any distributed algorithm. To robustify Jacobi and ESA to such faults, we modify them slightly by allowing each node to store its neighbors' most recent, successfully communicated values in its *local memory*. Let I^m be the set of active nodes at iteration m . A sleeping node $i \notin I^m$ updates neither its estimate nor its memory during iteration m (that is, $\hat{x}_i^m = \hat{x}_i^{m-1}$). Further, it does not communicate with its neighbors. A failing link between nodes j and i precludes node i from receiving any data from node j . In place of this missing value, node i recalls from its local memory the last successfully received value from node j . Let $\ell_{j \rightarrow i}(m)$ denote the iteration count corresponding to this most recently received value. In Robust ESA, active nodes $i \in I^m$ perform the following steps. (Robust Jacobi is defined similarly, since Jacobi is a particular case of ESA with singleton subgraphs.)

Robust ESA Update: Each node i attempts to exchange its estimate \hat{x}_i^{m-1} with all of its neighbors except those belonging to $s_k(i)$. If node i does not receive the transmitted \hat{x}_j^{m-1} (unsuccessful communication) from node j , then it recalls $\hat{x}_j^{\ell_{j \rightarrow i}(m)}$ from its memory. Node i updates its auxiliary variable \hat{y}_i^m according to (14) using either \hat{x}_j^{m-1} or $\hat{x}_j^{\ell_{j \rightarrow i}(m)}$ for $j \in \{\mathcal{N}(i) \setminus s_k(i)\}$.

Robust ESA Solve: Within the embedded subgraph $s_k(i)$, the nodes attempt to exchange their updated $\hat{y}_{s_k(i)}^m$. If node i does not receive the updated \hat{y}_j^m from $j \in s_k(i)$, then it recalls $\hat{y}_j^{\ell_{j \rightarrow i}(m)}$ from its memory. Node i solves for \hat{x}_i^m by inverting the matrix $V_{s_k(i)}$ according to (15).

In other words, nodes substitute the values not received due to unsuccessful communication with the most recently received values from memory and then perform the usual ESA update and solve steps from Section IV-A.

B. Robust Convergence

This subsection contains the paper's main theoretical contribution. Theorem 3 provides the general condition for Robust ESA's convergence in the presence of temporary faults. Theorem 4 identifies two specific scenarios in which Robust ESA converges; these are almost identical to the fault-free conditions in Theorem 2. See Appendix II for the proofs.

Below, $|J^{-1}K|$ denotes the matrix obtained by replacing each element in the matrix $J^{-1}K$ by its absolute value.

Theorem 3: Assume that during the Robust ESA update and Robust ESA solve steps of Section VI-A, the delays $d_{j \rightarrow i}(m) := m - \ell_{j \rightarrow i}(m)$ are bounded for all $(i, j) \in \mathcal{G}$ and that each node is updated within a finite number of iterations. If the ESA matrix splitting is such that $\rho(|J^{-1}K|) < 1$, then Robust ESA converges to the LMMSE estimate.

The condition $\rho(|J^{-1}K|) < 1$ is satisfied by the class of so-called *H-matrices* [15], [26], which are strictly *generalized* diagonally dominant [27]. That is, $J^{-1}K$ is an *H-matrix* of size N if and only if there exists an entrywise positive vector $u = (u_1, u_2, \dots, u_N)$ such that

$$|(J^{-1}K)_{ii}| u_i > \sum_{k \neq i} |(J^{-1}K)_{ik}| u_k, \quad \forall i = 1, 2, \dots, N$$

Further, note that the ESA splitting $V = J - K$ is such that if V is an *H-matrix*, then so is $J^{-1}K$ [15].

We can now identify two physical conditions for robust convergence that parallel the conditions in Theorem 2.

Theorem 4: Assume that link failures and node sleeping are temporary (as described in Theorem 3). If either of the following two conditions hold, then Robust ESA converges to the LMMSE estimate:

1. the partial correlations of x in (1) are nonnegative (that is, (16) holds);
2. the noise variance of ϵ in (1) is small

$$\mathbb{E}(\epsilon_i^2) \leq \frac{1}{(\max_i \Sigma_{ii}^{-1}) (\max_i \#\{\mathcal{N}(i)\})} \quad (19)$$

with $\max_i \Sigma_{ii}^{-1}$ the maximum value of the diagonal elements of Σ^{-1} , and $\max_i \#\{\mathcal{N}(i)\}$ the maximum number of neighbors for any node in the GM.

VII. EXPERIMENTS: FAULT TOLERANCE

We now investigate the robustness of Jacobi and ETA to sleeping nodes and communication failures. Since it is unclear how to modify embedded trees to handle such faults, we exclude it from this comparison. While loopy BP can be modified to function in the presence of faults, we excluded it from our experiments due to its poor performance in the fault-free case. The results presented in this section can be replicated using MATLAB code that can be downloaded from www.dsp.rice.edu/software.

A. Fault Generation

We generate the data as in Section V-A and use a simple independent model for fault generation. To study the effect of sleeping nodes, we assume that at each iteration each node independently goes to sleep with probability p and stays awake with probability $1 - p$. We model failing communication links similarly; at each iteration, each link can independently fail with probability p .

B. Comparison Metrics

We compare how Jacobi's and ETA's RNMSE decay rate (with respect to energy consumption) varies with fault probability. For each fault probability p , we have an RNMSE versus energy curve. For example, Fig. 4(a) and (b) displays the RNMSE versus energy λ curves for Jacobi and ETA at three different probabilities of failing links; as expected, the rate of RNMSE's decay decreases as the probability of failing links increases. To succinctly communicate both algorithm's performances over a range of probabilities, we will summarize each curve using a single quantity, namely, the decay rate.

When there are no failing links ($p = 0$), the RNMSE ^{m} in (18) can be upper-bounded by $c\rho^m$, with ρ the spectral radius of Theorem 1 and c a constant [13], [14]. Thus, we can expect the RNMSE versus energy curve for $p = 0$ to be log-linear. Interestingly, we observed that all RNMSE versus energy curves to be very close to log-linear; the trend exists for sleeping nodes

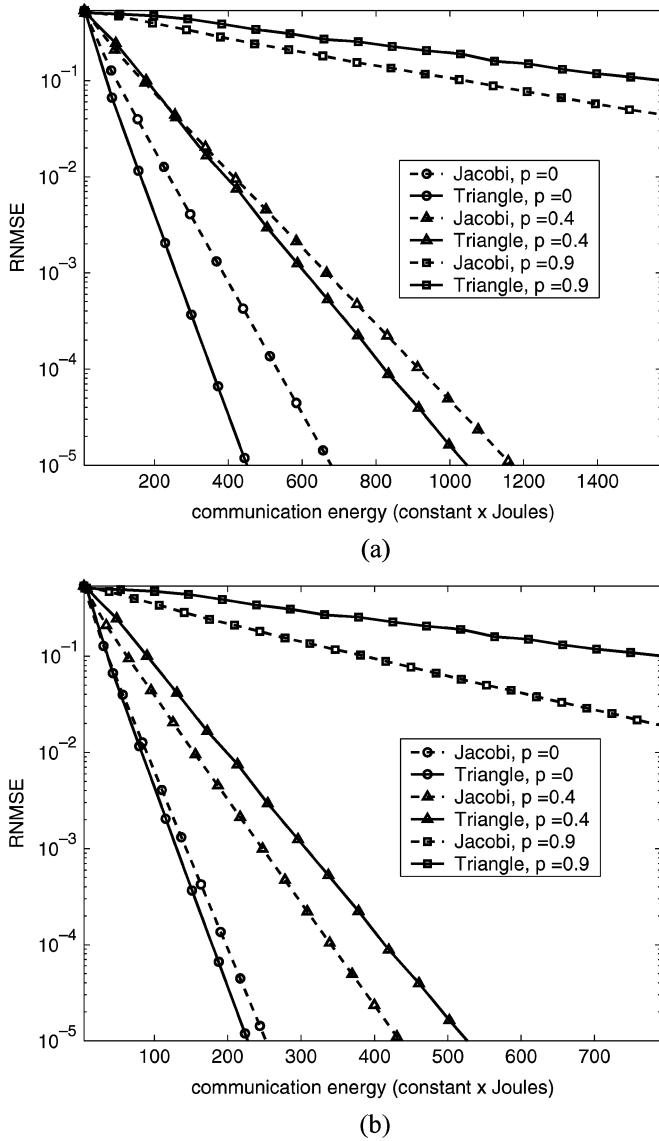


Fig. 4. Fault-prone (failing links) case. Simulation results for distributed estimation with the sensor network from Fig. 1 using Jacobi and ETA. The communication links failed at each transmission with probability p . (a) RNMSE versus energy (unidirectional communication); (b) RNMSE versus energy (omnidirectional communication).

as well. Based on this observation, for any fault probability p , we model

$$\begin{aligned} \text{RNMSE}_p(\lambda) &\approx \gamma e^{-\beta_p \lambda} \\ \log(\text{RNMSE}_p(\lambda)) &\approx -\beta_p \lambda + \log(\gamma) \end{aligned} \quad (20)$$

with λ the consumed energy and γ the initial error at the start of the iterations. In our experiments, all of the methods were initialized with the null vector $\hat{x}^0 = 0$. Hence, all techniques have the same RNMSE at startup. This means that γ is invariant to the algorithm as well as to the fault probability. The β_p encapsulates the RNMSE rate of decay at fault probability p ; the larger the β_p , the faster the RNMSE decay.

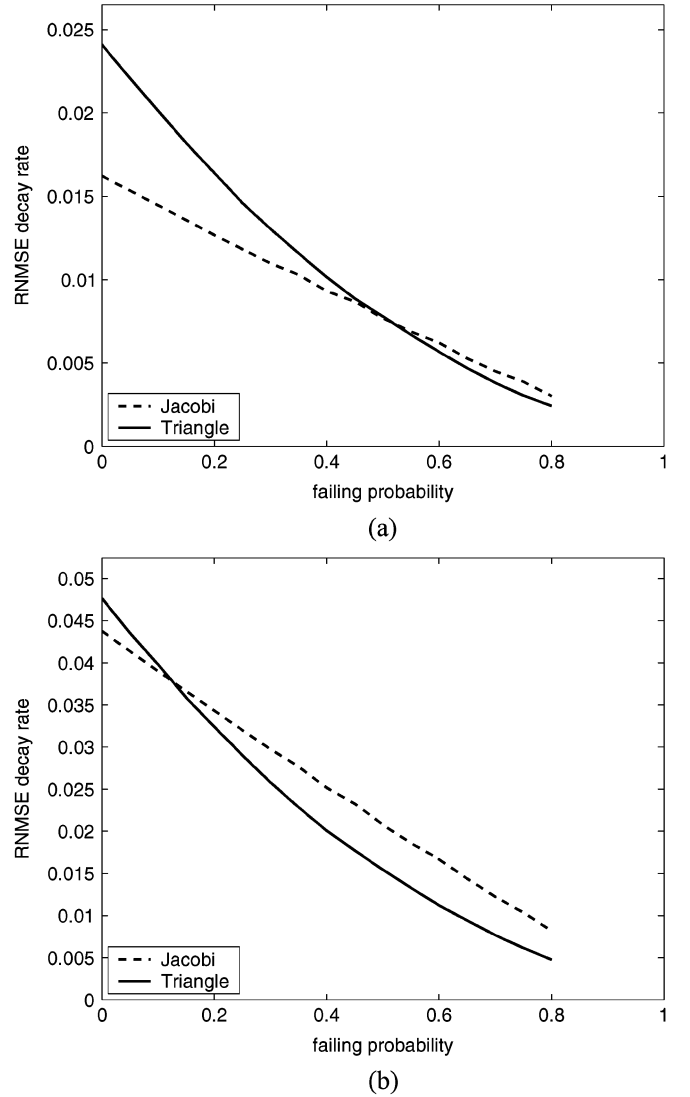


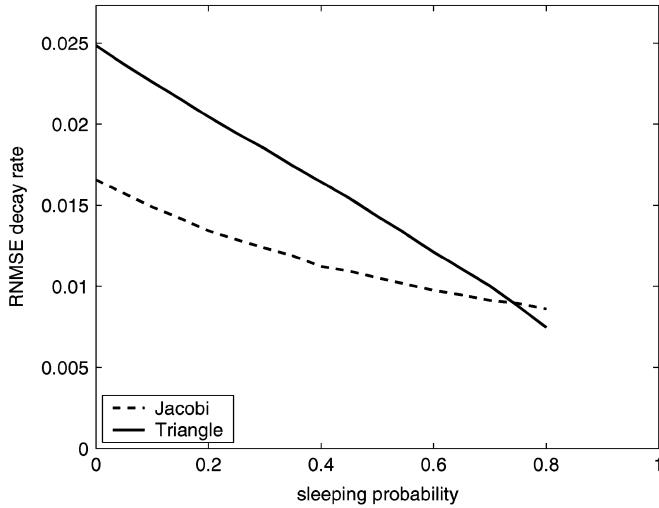
Fig. 5. Fault-prone (failing links) case. Evolution of the RNMSE decay rate (β_p) with increasing probability p of communication link failures. The convergence rate deteriorates as failures occur more often. (a) RNMSE decay rate versus failing probability (unidirectional communication); (b) RNMSE decay rate versus failing probability (omnidirectional communication).

Using the least-squares fit β_p , the model (20) achieves an R^2 fit of $1 - O(10^{-5})$ for the curves in Fig. 4.⁹ Henceforth, we summarize the performance of both ETA and Jacobi by plotting the rate of RNMSE decay β_p versus the fault probability p , where the least-squares fit β_p is computed using the respective RNMSE versus energy curve at each probability p .

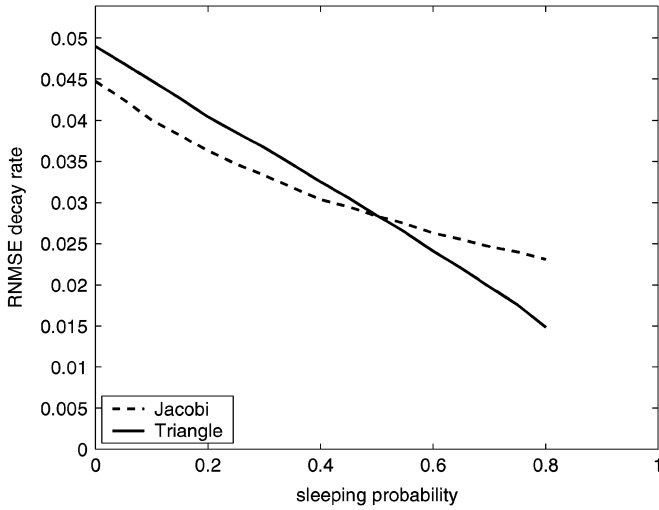
C. Fault-Prone Results: Failing Links

We now study the effect of the link failure probability on the performance of Robust Jacobi and Robust ETA. Fig. 5(a) and (b) illustrates the RNMSE decay rate versus failing link probability curves for unidirectional and omnidirectional transmitter scenarios, respectively.

⁹The R^2 statistic $R^2 \in [0, 1]$ is the ratio between the variance explained by the model and the total variation the observations. $R^2 = 1$ means that the model fits the data perfectly [28].



(a)

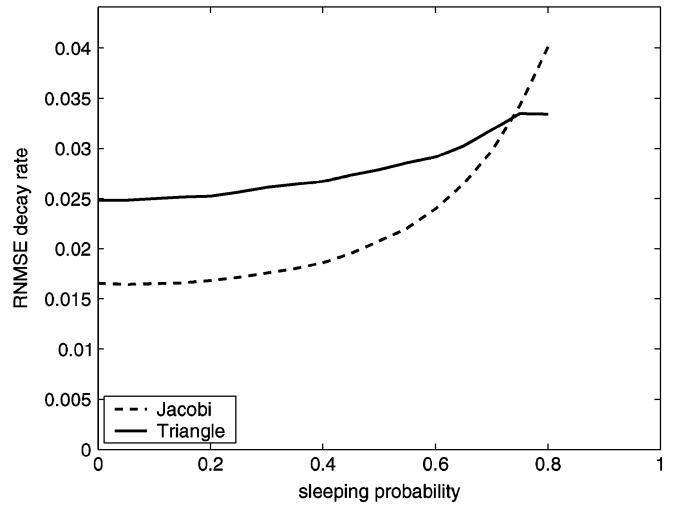


(b)

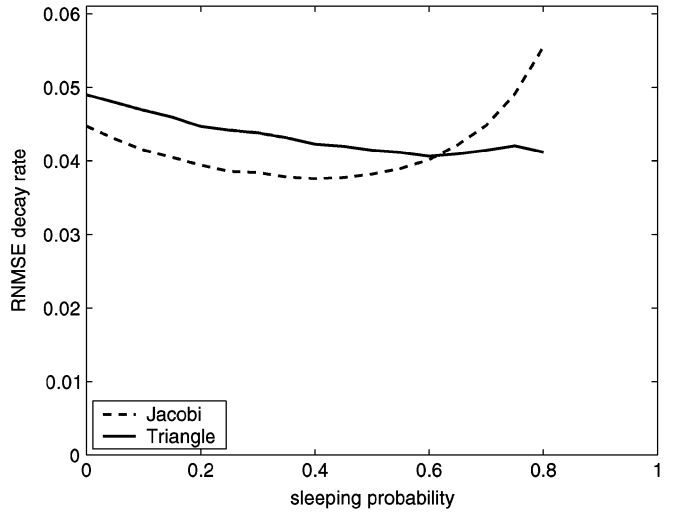
Fig. 6. Fault-prone (unsynchronized sleeping) case. Evolution of the RNMSE decay rate (β_p) with increasing probability p of node sleeping. Nodes do not know the sleep patterns of their neighbors. (a) RNMSE decay rate versus failing probability (unidirectional communication); (b) RNMSE decay rate versus failing probability (omnidirectional communication).

In the unidirectional case [see Fig. 5(a)], Robust ETA enjoys faster RNMSE decay rates than Robust Jacobi for smaller probabilities of failing links (up to $p \approx 0.5$); the performances are comparable at larger fault probabilities. We observe that Robust ETA's relative degradation in performance is larger than Robust Jacobi as more links fail. The reason is that Robust ETA requires internode communications for *both* the Robust update step and the Robust solve step. Thus, each node requires two successful back-to-back communications to effectively update its LMMSE estimate. At larger probabilities of failing links, this becomes rarer, resulting in performance degradation. In contrast, during Robust Jacobi iterations, each node needs only one successful communication (during its update step) to effectively update its LMMSE estimate.

In the omnidirectional case [see Fig. 5(a)], Robust ETA enjoys faster RNMSE decay rates than Robust Jacobi only for very low probabilities of failing links (up to $p \approx 0.15$). This is be-



(a)



(b)

Fig. 7. Fault-prone (synchronized sleeping) case. Evolution of the RNMSE decay rate β_p with increasing probability p of node sleeping. Each node knows the sleep patterns of its neighbors. (a) RNMSE decay rate versus failing probability (unidirectional communication); (b) RNMSE decay rate versus failing probability (omnidirectional communication).

cause Jacobi is perfectly suited to exploit omnidirectional transmitters (see Section V-C).

An interesting approach would be to adaptively switch between the Robust Jacobi and Robust ETA (or larger subgraph decompositions) based on the respective RNMSE decay rates when the fault probability is known. Fig. 5(a) and (b) indicates that, given the benchmark performances of Robust Jacobi and Robust ETA at some failing link probability, their performance at other probabilities are predictable. In fact, we have verified that the ETA's RNMSE decay rate is approximately quadratic in fault probability p , whereas Jacobi's RNMSE decay rate is approximately linear in p . Note that at $p = 1$, both decay rates should be zero (no convergence).

D. Fault-Prone Results: Sleeping Nodes

We consider two sleeping scenarios. In the first scenario (see Fig. 6), we assume that a node's neighbors are unaware of its

sleeping pattern. Consequently, its neighbors will waste energy trying to update it; we term this *unsynchronized sleeping*. In the second scenario (see Fig. 7), we study the benefits of *synchronized sleeping*, where a node's neighbors are aware of its sleeping pattern and can conserve energy.

For the unsynchronized sleeping scenario [see Fig. 6(a) and (b)], similar to the failing link results in Section VII-C, ETA enjoys faster RNMSE decay rates than Jacobi for smaller sleeping probabilities; at larger probabilities, Jacobi's performance is marginally better.

The degradation in decay rates could have some surprising implications on sensor network design. Periodic sleeping has been widely touted as an effective and essential strategy for nodes to conserve energy [23], [24]. However, Fig. 6(a) and (b) reminds us that unsynchronized sleeping could force sensors to consume significant additional energy to achieve a target error tolerance in LMMSE estimation.

Synchronized sleeping fares much better. Fig. 7(a) and (b) illustrates the RNMSE decay rate versus sleeping probability curves for synchronized sleeping. Interestingly, the RNMSE decay rate remains surprisingly stable (with a marginal *increase* in the unidirectional case) at higher probabilities of sleeping [see Fig. 7(a)]. The cause for this phenomenon is that, at higher sleeping probabilities, nodes update their values almost sequentially with the latest values from their neighbors. This sequential Gauss–Seidel aspect enhances the convergence of the respective algorithms with respect to the *energy* spent. Obviously, the *time* to convergence increases with increasing sleeping probability.

VIII. DISCUSSION AND CONCLUSION

The embedded subgraphs algorithm (ESA) is a new iterative, distributed LMMSE estimation scheme for loopy Gaussian graphical models that is based on the block Jacobi matrix splitting algorithm [13], [14]. ESA decomposes a GM into disjoint subgraphs, performs LMMSE estimation on each subgraph, and then updates this estimate by collaborating with neighboring nodes and subgraphs.

We have demonstrated that ESA can be robustified to faults such as sleeping nodes and failing links. One of our key theoretical contributions is a general sufficient condition for ESA's convergence in the presence of temporary faults. Further, we identify two specific cases—when the noise variance is small and when the original signal's partial correlations are nonnegative—for which ESA is guaranteed to converge (assuming temporary faults).

Our experimental results provide some interesting insights to the tradeoff between a sensor network's communication energy consumption and iterative algorithm convergence. Conventional wisdom says that sensor nodes should be powered down periodically to conserve energy [23], [24]. However, our results indicate that if the sensors are switched off in an uncoordinated fashion during the iterations, then the network could end up consuming significant additional energy to achieve a specified estimate error tolerance. Moreover, our experiments indicate that if nodes synchronize their sleep and avoid any waste of energy, then the RNMSE rate of decay with respect to the energy con-

sumed *increases* with larger probability of failure. Naturally, the time to convergence increases with probability of failure.

We envision many opportunities for future research and performance enhancements. Currently, we are trying to understand how to best cycle through a range of different embedded subgraph decompositions in order to speed up convergence (by analogy to [7]). We are also investigating the optimal size of the embedded subgraphs. As their size increases, the amount of residual error decay per iteration increases, but so does their sensitivity to sleeping and communication faults. Further, we are studying algorithms to estimate a GM (correlation structure) for a sensor network from sample data. An expectation-maximization (EM)-type approach [25] could prove useful to address this issue.

Other interesting avenues include the design of an efficient variance estimation algorithm, extensions to track changes over time (Kalman filtering), studying the effect of quantization of the parameters and estimates, and optimizing interaction between distributed estimation schemes and ad hoc wireless network routing protocols. The more general problem of finding a consensus amongst sensors in a decentralized manner has been studied in different settings such as averaging [29]–[31], hypothesis testing [32] and source localization [33]. For such problems, it would be interesting to study the convergence of the proposed solutions as the number of temporary faults increases in the network.

APPENDIX I PROOF OF THEOREM 2

A. Nonnegative Partial Correlation Condition

Proof: Below, we use the notation $U > 0$ (or $U \geq 0$) when a matrix U has all positive (or nonnegative) elements. First, we show that V is an M -matrix; that is, V is nonsingular and its inverse V^{-1} has all positive elements [13, p. 85]. Since the partial correlations are nonnegative, the off-diagonal elements of V are all ≤ 0 (see Section II-A). Further, V is symmetric and positive definite. Hence, from [13, p. 85, Corollary 3], $V^{-1} \geq 0$, thereby proving that V is an M -matrix. Second, by [13, p. 85, Th. 3.12], J is also an M -matrix. Hence, $J^{-1} > 0$ and $J^{-1}K \geq 0$ since by assumption $K \geq 0$. The splitting $V = J - K$ is then termed *regular*. For regular splitting, [13, p. 85, Th. 3.13] tells us that if $V^{-1} \geq 0$, then the spectral radius $\rho(J^{-1}K) < 1$.

B. Small Noise Variance Condition

Proof: In this case, we will invoke a result from [34] and then verify that $J + K$ is positive definite when (17) holds.

Theorem 5: [34] Let $V = J - K$ be a splitting of the Hermitian matrix V such that $J^* + K$, ($*$ denotes conjugate transpose) is positive definite. Then $\rho(J^{-1}K) < 1$ if and only if V is positive definite.

The V in (2) is clearly a positive definite, Hermitian (it is real and symmetric) matrix. Further, the matrix J is symmetric and real for both ESA and Jacobi. Hence, ESA and Jacobi will converge if $J + K$ is positive definite.

Using (5) and (3), we have

$$J + K = V + 2K = \Sigma^{-1} + R^{-1} + 2K. \quad (21)$$

Since the observations x are not linearly dependent, Σ^{-1} is positive definite. We now need to ensure that $R^{-1} + 2K$ is positive definite. The matrix R^{-1} is diagonal with entries $1/\mathbb{E}(\epsilon_i^2)$. The eigenvalues of K , which is Hermitian, are real. Let η denote the maximum absolute eigenvalue of K . For any $u \in \mathbb{C}^N$,

$$u^H(R^{-1} + 2K)u \geq \min_i \left(\frac{1}{\mathbb{E}(\epsilon_i^2)} \right) \|u\|_2^2 - 2\eta \|u\|_2^2 \geq 0$$

assuming (17) holds. Using Theorem 5, since $J + K$ is positive definite (see (21)), $\rho(J^{-1}K) < 1$. Theorem 1 then guarantees convergence. \square

APPENDIX II ROBUST CONVERGENCE

A. Proof of Theorem 3

Proof: We begin by showing that iterations of the Robust ESA described in Section VI-A are *asynchronous iterations* [15]. As in Section VI-A, let I^m be the set of active (non-sleeping) nodes at iteration m , and let $\ell_{j \rightarrow i}(m)$ denote the iteration count corresponding to the most recent value that sensor i successfully received from j . At iteration m , an active sensor $i \in I^m$ will either receive from each sensor $j \in \mathcal{N}(i)$ the values \hat{x}_j^{m-1} (working link) or it will recall from its memory the last value received at the iteration count $\ell_{j \rightarrow i}(m)$, $\ell_{j \rightarrow i}(m) < m - 1$ (failing link). This scenario has been well studied in parallel computing.

Definition 6 (Asynchronous Iteration): For $m \in \mathbb{N}$, let $I^m \subseteq \{1, 2, \dots, n\}$ and the iteration counts $\ell_{j \rightarrow i}(m) \in \mathbb{N}_0$, $(i, j) \in \mathcal{G} = (\mathcal{E}, \mathcal{V})$ be such that

- (A) $\ell_{j \rightarrow i}(m) \leq m - 1$, for $i, j \in \{1, 2, \dots, N\}$
- (B) $\lim_{m \rightarrow \infty} \ell_{j \rightarrow i}(m) = \infty$, for $i, j \in \{1, 2, \dots, N\}$
- (C) $\#\{m \in \mathbb{N} | i \in I^m\} = \infty$, for $i \in \{1, 2, \dots, N\}$

with $\#(\cdot)$ denoting the number of elements in a set. Given an initial guess \hat{x}^0 and the above three conditions, the following iteration is termed *asynchronous*

$$\hat{x}_i^m = \begin{cases} \hat{x}_i^{m-1}, & \text{for } i \notin I^m \\ H_i \left(\hat{x}_1^{\ell_{1 \rightarrow i}(m)}, \dots, \hat{x}_n^{\ell_{n \rightarrow i}(m)} \right), & \text{for } i \in I^m \end{cases} \quad (22)$$

where H_i is the i th component of an iteration operator H .

The first condition (A) on $\ell_{j \rightarrow i}(m)$ requires that only values computed earlier are used in the current approximation \hat{x}_j^m of the solution. The second condition (B) requires that as the algorithm proceeds, new information is continually transmitted to each sensor. The third condition (C) stipulates that a sensor cannot sleep forever.

The Robust ESA update and solve steps obviously satisfy condition (A). The assumption requiring bounded delays $d_{j \rightarrow i}(m) = m - \ell_{j \rightarrow i}(m)$ implies condition (B). The condition on each node being updated within a finite number of iterations effectively means that a node cannot sleep forever, and this is given by (C). Hence, under the assumptions of Theorem 3, the Robust ESA update and solve steps in Section VI-A constitute an asynchronous iteration.

When solving the linear system $Vx = \bar{y}$ from (2) using a matrix splitting $V = J - K$, the iteration operator H is given by

$$H : \mathbb{R}^n \rightarrow \mathbb{R}^n, x \rightarrow J^{-1}(Kx + \bar{y}). \quad (23)$$

Theorem 4.1 from [15] states that if the spectral radius $\rho(|H|) = \rho(|J^{-1}K|) < 1$, then the asynchronous iteration (22) will converge to \hat{x} , the solution of $Vx = \bar{y}$. Hence, the result follows. \square

B. Proof of Theorem 4

1) *Nonnegative Partial Correlation Condition:* From Appendix I-A, V is an M -matrix. Since M -matrices are a subclass of H -matrices [15], [27], the result holds.

2) *Small Noise Variance Condition:* We only need to show that V is an H -matrix. Indeed, for ESA-type splitting $V = J - K$, the matrix $J^{-1}K$ is an H -matrix as soon as V is an H -matrix [15].

To show that V is an H -matrix, we need to find a positive vector u such that

$$|V_{ii}|u_i > \sum_{k \neq i} |V_{ik}|u_k, \quad \forall i. \quad (24)$$

In our model, $V = \Sigma^{-1} + (1/\sigma^2)$, where $\sigma^2 = \mathbb{E}(\epsilon^2)$ and 1 denotes the identity matrix. The entries Σ_{ik}^{-1} and the partial correlation coefficients are related by [3, p. 131]

$$\Sigma_{ik}^{-1} = -\text{Corr}(x_i, x_k | x_{\mathcal{V} \setminus \{i, k\}}) \sqrt{\Sigma_{ii}^{-1} \Sigma_{kk}^{-1}}.$$

Thus, (24) can be rewritten as

$$\left| \Sigma_{ii}^{-1} + \frac{1}{\sigma^2} \right| u_i > \sum_{k=1, k \neq i}^N \left| -\text{Corr}(x_i, x_k | x_{\mathcal{V} \setminus \{i, k\}}) \right| \times \sqrt{\Sigma_{ii}^{-1} \Sigma_{kk}^{-1}} u_k. \quad (25)$$

We now exploit two observations. First, $\text{Corr}(x_i, x_k | x_{\mathcal{V} \setminus \{i, k\}})$ is nonzero only when nodes i and k are neighbors. So the sum in (25) can be replaced by the sum over $\mathcal{N}(i)$. Second, $\text{Corr}(x_i, x_k | x_{\mathcal{V} \setminus \{i, k\}})$ lies between -1 and 1 . Hence, we seek a positive vector u such that

$$\left| \sqrt{\Sigma_{ii}^{-1}} \left(1 + \frac{1}{\sigma^2 \Sigma_{ii}^{-1}} \right) \right| u_i > \sum_{k \in \mathcal{N}(i)} \sqrt{\Sigma_{kk}^{-1}} u_k. \quad (26)$$

Assuming that (19) holds, it is easy to verify that (26) is satisfied when $u_k = \left(\sqrt{\Sigma_{kk}^{-1}} \right)^{-1}$; note that $\Sigma_{kk}^{-1} = \mathbb{E}(x_i^2 | x_{\mathcal{V} \setminus i})^{-1} > 0$ [3, p. 129].

ACKNOWLEDGMENT

The authors thank D. Sorensen for several fruitful discussions and valuable pointers. The authors also thank the anonymous reviewers and the editor for their valuable comments and suggestions.

REFERENCES

- [1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proc. ACM/IEEE MobiCom '99*, Aug. 1999, pp. 263–270.
- [2] A. Wang and A. Chandrakasan, "Energy-efficient DSP's for wireless sensor networks," *IEEE Signal Proces. Mag.*, vol. 19, no. 4, pp. 68–78, Jul. 2002.
- [3] S. L. Lauritzen, *Graphical Models*. Oxford, U.K.: Oxford Univ. Press, 1996.
- [4] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.
- [5] F. Araújo and L. Rodrigues, "Fast localized delaunay triangulation," presented at the 8th Int. Conf. Principles Distributed Systems (OPODIS), Grenoble, France, Dec. 2004.
- [6] R. Wagner, H. Choi, R. Baraniuk, and V. Delouille, "Distributed wavelet transform for irregular sensor network grids," presented at the IEEE Statistical Signal Processing Workshop, Bordeaux, France, Jul. 2005.
- [7] E. Sudderth, M. J. Wainwright, and A. S. Willsky, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," *IEEE Trans. Signal Process.*, vol. 52, no. 11, pp. 3136–3150, Nov. 2004.
- [8] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [9] K. Piarre and P. R. Kumar, "Extended message passing algorithm for inference in loopy Gaussian graphical models," *Ad Hoc Netw.*, vol. 2, pp. 153–169, 2004.
- [10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
- [11] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, pp. 2173–2200, 2001.
- [12] M. J. Wainwright, "Stochastic processes on graphs with cycles: Geometric and variational approaches," Ph.D. dissertation, ECE Dept., MIT, Cambridge, MA, Jan. 2002.
- [13] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1962.
- [14] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS, 1996.
- [15] A. Frommer and B. Szyld, "On asynchronous iterations," *J. Comput. Appl. Math.*, vol. 123, pp. 201–216, 2000.
- [16] R. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Trans. Inf. Theory*, vol. 29, no. 1, pp. 60–72, Jan. 1983.
- [17] T. Speed and H. Kiiveri, "Gaussian Markov distributions over finite graphs," *Ann. Stat.*, vol. 14, no. 138–150, 1986.
- [18] E. Sudderth, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," Master's thesis, ECE Dept., MIT, Cambridge, MA, 2002.
- [19] A. Abdalla, N. Deo, and P. Gupta, "Random-tree diameter and diameter-constrained MST," *Congressus Numerantium*, vol. 144, pp. 161–182, 2000.
- [20] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," in *Proc. 3rd Int. Symp. Information Processing Sensor Networks (IPSN)*, Berkeley, CA, Apr. 2004, pp. 405–413.
- [21] V. Delouille, R. Neelamani, V. Chandrasekaran, and R. G. Baraniuk, "The embedded triangles algorithm for distributed estimation in sensor networks," in *Proc. IEEE Statistical Signal Processing Workshop*, St. Louis, MO, 2003, pp. 357–360.
- [22] R. Min, M. Bhardwaj, S. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. P. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," *IEEE Wireless Commun.*, vol. 9, no. 4, pp. 28–39, Aug. 2002.
- [23] J.-F. Chamberland and V. Veeravalli, "The art of sleeping in wireless sensing systems," in *Proc. IEEE Statistical Signal Processing Workshop*, St. Louis, MO, 2003, pp. 9–12.
- [24] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology Control protocols to conserve energy in wireless ad hoc networks UCLA, Los Angeles, CA, Tech. Rep. 6, Jan. 2003.
- [25] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*. New York: Springer, 1999.
- [26] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia, PA: SIAM, 1994.
- [27] B. Li, L. Li, M. Harada, H. Niki, and M. Tsatsomeros, "An iterative criterion for H-matrices," *Lin. Alg. Its Appl.*, vol. 271, pp. 179–190, 1998.
- [28] S. Chatterjee and A. Hadi, "Influential observations, high leverage points, and outliers in linear regression," *Stat. Sci.*, pp. 379–416, 1986.
- [29] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," presented at the 4th Int. Symp. Information Processing Sensor Networks (IPSN), Los Angeles, CA, Apr. 25–27, 2005.
- [30] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.
- [31] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," *Proc. IEEE INFOCOM*, vol. 3, pp. 1653–1664, Mar. 2005.
- [32] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed Bayesian hypothesis testing in sensor networks," in *Proc. Amer. Control Conf.*, Boston, MA, 2004, vol. 6, pp. 5369–5374.
- [33] M. Rabbat, R. Nowak, and J. Bucklew, "Robust decentralized source localization via averaging," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 5, pp. v/1057–v/1060, Mar. 2005.
- [34] V. Migallón and J. Penadés, "Convergence of two-stage iterative methods for Hermitian positive definite matrices," *App. Math. Lett.*, vol. 10, no. 3, pp. 79–83, 1997.



Véronique Delouille (M'05) received the B.Tech. degree in applied mathematics from the Catholic University of Louvain in 1997, the M.S. degree in statistics from the University of Kent at Canterbury in 1998, and the Ph.D. degree in statistics from the Catholic University of Louvain in 2002.

After spending 2003 to 2004 in the Digital Signal Processing Group at Rice University, Houston, TX, she joined the Solar Physics Department of the Royal Observatory of Belgium, Brussels. Her research interests are in statistics and signal processing, include

wavelet regression estimation, sensor networks, and multiscale analysis of solar images.

Dr. Delouille is a member of the Belgian Statistical Society and the SOHO, PROBA2, and SO solar observation missions.



Ramesh "Neelsh" Neelamani (S'98–M'03) received the B.Tech. degree from the Indian Institute of Technology—Bombay in 1997 and the M.S. and Ph.D. degrees from Rice University, Houston, TX, in 1999 and 2003, respectively, all in electrical engineering.

He is currently a Research Scientist at Exxon-Mobil's Upstream Research Company, Houston, TX. His research interests include signal and image processing topics such as multiscale signal representations and algorithms, statistical signal processing, color processing, distributed algorithms, and pattern recognition, and most recently has developed an interest in discovering all forms of dinosaur fossils.

Dr. Neelamani won the second prize in the IEEE All-India Student Paper Contest in 1995. He is a member of Society of Exploration Geophysicists (SEG) and the Society for Industrial and Applied Mathematics (SIAM).



Richard G. Baraniuk (S'85–M'93–SM'98–F'02) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1987, the M.Sc. degree from the University of Wisconsin–Madison in 1988, and the Ph.D. degree from the University of Illinois at Urbana–Champaign in 1992, all in electrical engineering.

After spending 1992 to 1993 at Ecole Normale Supérieure, Lyon, France, he joined Rice University, Houston, TX, where he is currently the Victor E. Cameron Professor of Electrical and Computer

Engineering and Director of the Connexions Project. He spent sabbaticals at Ecole Nationale Supérieure de Télécommunications, Paris, France, in 2001 and Ecole Fédérale Polytechnique de Lausanne, Switzerland, in 2002. His research interests in signal and image processing include wavelets and multiscale analysis, statistical modeling, and sensor networks.

Dr. Baraniuk received a NATO postdoctoral fellowship from NSERC in 1992, the National Young Investigator award from NSF in 1994, a Young Investigator Award from ONR in 1995, the Rosenbaum Fellowship from the Isaac Newton Institute of Cambridge University in 1998, the C. Holmes MacDonald National Outstanding Teaching Award from Eta Kappa Nu in 1999, the Charles Duncan Junior Faculty Achievement Award from Rice in 2000, the ECE Young Alumni Achievement Award from the University of Illinois in 2000, and the George R. Brown Award for Superior Teaching at Rice University in 2001, 2003, and 2006.