Title

Distributed optimal estimation from relative measurements for localization and time synchronization

Journal

DISTRIBUTED COMPUTING IN SENSOR SYSTEMS, PROCEEDINGS, 4026

ISSN

0302-9743

Authors

Barooah, P
da Silva, N M
Hespanha, Joao P

Publication Date

2006

Peer reviewed

Distributed Optimal Estimation from Relative Measurements for Localization and Time Synchronization

Prabir Barooah[1], Neimar Machado da Silva[2], and João P. Hespanha[1]

[1] University of California, Santa Barbara, CA 93106, USA
{pbarooah, hespanha}@ece.ucsb.edu
[2] Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
neimarms@gmail.com

Abstract. We consider the problem of estimating vector-valued variables from noisy "relative" measurements. The measurement model can be expressed in terms of a graph, whose nodes correspond to the variables being estimated and the edges to noisy measurements of the difference between the two variables. This type of measurement model appears in several sensor network problems, such as sensor localization and time synchronization. We consider the optimal estimate for the unknown variables obtained by applying the classical Best Linear Unbiased Estimator, which achieves the minimum variance among all linear unbiased estimators.

We propose a new algorithm to compute the optimal estimate in an iterative manner, the *Overlapping Subgraph Estimator* algorithm. The algorithm is distributed, asynchronous, robust to temporary communication failures, and is guaranteed to converges to the optimal estimate even with temporary communication failures. Simulations for a realistic example show that the algorithm can reduce energy consumption by a factor of two compared to previous algorithms, while achieving the same accuracy.

1 Introduction

We consider an estimation problem that is relevant to a large number of sensor networks applications, such as localization and time synchronization. Consider n vector-valued variables $x_1, x_2, \ldots, x_n \in \mathbb{R}^k$, called node variables, one or more of which are known, and the rest are unknown. A number of noisy measurements of the difference between certain pairs of these variables are available. We can associate the variables with the nodes $\mathbf{V} = \{1, 2, \ldots, n\}$ of a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and the measurements with the edges $\mathbf{E}$ of it, consisting of ordered pairs (u, v) such that a noisy "relative" measurement between x_u and x_v is available:

$$\zeta_{uv} = x_u - x_v + \epsilon_{uv}, \tag{1}$$

P. Gibbons et al. (Eds.): DCOSS 2006, LNCS 4026, pp. 266–281, 2006.

where the ϵ_{uv}'s are uncorrelated zero-mean noise vectors with known covariance matrices. That is, for every edge $e \in \mathbf{E}$, $P_e = \mathrm{E}[\epsilon_e \epsilon_e^T]$ is known, and $\mathrm{E}[\epsilon_e \epsilon_{\bar{e}}^T] = 0$ if $e \neq \bar{e}$. The problem is to estimate all the unknown node variables from the measurements. We call $\mathbf{G}$ a *measurement graph* and x_u the *u-th node variable*. The node variables that are known are called the *reference variables* and the corresponding nodes are called the *reference nodes*. The relationship of this estimation problem with sensor network applications is discussed in section 1.1.

Our objective is to construct an optimal estimate $\hat{x}_u^*$ of x_u for every node $u \in \mathbf{V}$ for which x_u is unknown. The *optimal estimate* refers to the estimate produced by the classical Best Linear Unbiased Estimator (BLUE), which achieves the minimum variance among all linear unbiased estimators [1]. To compute the optimal estimate directly one would need all the measurements and the topology of the graph (cf. section 2). Thus, if a central processor has to compute the $\hat{x}_u^*$s, all this information has to be transmitted to it. In a large ad-hoc network, this burdens nodes close to the central processor more than others. Moreover, centralized processing is less robust to dynamic changes in network topology resulting from link and node failures. Therefore a distributed algorithm that can compute the optimal estimate while using only local communication will be advantageous in terms of scalability, robustness and network life.

In this paper we propose a new distributed algorithm, which we call the *Overlapping Subgraph Estimator* (OSE) algorithm, to compute the optimal estimates of the node variables in an iterative manner. The algorithm is distributed in the sense that each node computes its own estimate and the information required to perform this computation is obtained from communication with its one-hop neighbors. We show that the proposed algorithm is correct (i.e., the estimates converge to the optimal estimates) even in the presence of faulty communication links, as long as certain mild conditions are satisfied. The OSE algorithm asymptotically obtains the optimal estimate while simultaneously being scalable, asynchronous, distributed and robust to communication failures.

1.1 Motivation and Related Work

Optimal Estimation. The estimation problem considered in this paper is motivated by sensornet applications such as time synchronization and location estimation. We now briefly discuss these applications.

In a network of sensors with local clocks that progress at the same rate but have unknown offsets between them, it is desirable to estimate these offsets. Two nodes u and v can obtain a measurement of the difference between their local times by exchanging time stamped messages. The resulting measurement of clock offsets can be modeled by (1)(see [2] for details). The problem of estimating the offset of every clock with respect to a single reference clock is then a special case of the problem considered in this paper. Karp *et. al.* [3] have also investigated this particular problem. The measurement model used in [3] can be seen as an alternative form of (1). In this application, the node variable x_u is the offset of u's local time with respect to a "reference" clock, and is a scalar variable.

Optimal Estimation from relative measurements with *vector-valued* variables was investigated in [4, 5]. Localization from range and bearing measurements is an important sensor network application that can be formulated as a special case of the estimation problem considered in this paper. Imagine a sensor network where the nodes are equipped with range and bearing measurement capability. When the sensors are equipped with compasses, relative range and bearing measurement between two nodes can be converted to a measurement of their relative position vector in a global Cartesian reference frame. The measurements are now in the form (1), and the optimal location estimates for the nodes can now be computed from these measurements (described in section 2). In this application the node variables are vectors.

Several localization algorithms have been designed assuming only relative range information, and a few, assuming only relative angle measurement. In recent times combining both range and bearing information has received some attention [6]. However, to the best of our knowledge, no one has looked at the localization problem in terms of the noisy measurement model (1). The advantage of this formulation is that the effect of measurement noise can be explicitly accounted for and filtered out to the maximum extent possible by employing the classical Best Linear Unbiased Estimator(BLUE). This estimator produces the minimum variance estimate, and hence is the most accurate on average. Location estimation techniques using only range measurement can be highly sensitive to measurement noises, which may introduce significant errors into the location estimate due to flip ambiguities [7]. The advantage of posing the localization problem as an estimation problem in Cartesian coordinates using the measurement model (1) is that the *optimal* (minimum variance) estimates all node positions in a connected network can be unambiguously determined when only one node that knows its position. A large number of well placed beacon nodes that know their position and broadcast that to the network – a usual requirement for many localization schemes – are not required.

Distributed Computation. Karp *et. al.* [3] considered the optimal estimation problem for time synchronization with measurements of pairwise clock offsets, and alluded to a possible distributed computation of the estimate, but stopped short of investigating it. In [5], we have proposed a distributed algorithm for computing the optimal estimates of the node variables that was based on the Jacobi iterative method of solving a system of linear equations. This Jacobi algorithm is briefly discussed in section 2. Although simple, robust and scalable, the Jacobi algorithm proposed in [5] suffered from a slow convergence rate. The OSE algorithm presented in this paper has a much faster convergence rate than the Jacobi algorithm. Delouille *et. al.* [8] considered the minimum mean squared error estimate of a different problem, in which absolute measurements of random node variables (such as temperature) were available, but the node variables were correlated. They proposed an Embedded Polygon Algorithm (EPA) for computing the minimum mean squared error estimate of node variables in a distributed manner, which was essentially a block-Jacobi iterative method for solving a set

of linear equations. Although the problem in [8] was quite different from the problem investigated in this paper, their EPA algorithm could be adapted to apply to our problem. We will call it the *modified EPA*. Simulations show that the OSE algorithm converges faster than the modified EPA.

Energy Savings. Since OSE converges faster, it requires fewer iterations for the same estimation error, which leads to less communication and hence saves energy in ad-hoc wireless networks. Here estimation error refers to the difference between the optimal estimate and the estimate produced by the algorithm. It is critical to keep energy consumption at every node at a minimum, since battery life of nodes usually determines useful life of the network. The improved performance of OSE comes from the nodes sending and processing larger amounts of data compared to Jacobi and modified EPA. However, the energy cost of sending additional data can be negligible due to the complex dependence of energy consumption in wireless communication on radio hardware, underlying PHY and MAC layer protocols, network topology and a host of other factors.

Investigation into energy consumption of wireless sensor nodes has been rather limited. Still, we can get an idea of which parameters are important for energy consumption from the studies reported in [9, 10, 11]. It is reported in [9] that for very short packets (in the order of 100 bits), transceiver startup dominates the power consumption; so sending a very short message offers no advantage in terms of energy consumption over sending a somewhat longer message. In fact, in a recent study of dense network of IEEE 802.15.4 wireless sensor nodes, it is reported in transmitted energy per bit in a packet *decreases monotonically* upto the maximum payload [10]. One of the main findings in [11] was that in highly contentious networks, "transmitting large payloads is more energy efficient". On the other hand, receive and idle mode operation of the radio is seen to consume as much energy as the transmit mode, if not more [12]. Thus, the number of packets (sent and received) appear to be a better measure to predict energy consumption than the number of bits.

In light of the above discussion, we used the number of packets transmitted and received as a coarse measure of the energy consumed by a node during communication. With number of packets as the energy consumption metric, simulations indicate that the OSE algorithm can cut down the average energy consumption for a given estimation accuracy as much by a factor of two or more (compared to Jacobi and modified EPA).

1.2 Organization

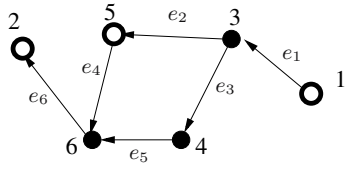
The paper is organized as follows. In section 2, the optimal estimator for the problem at hand is described. In section 3, we describe three algorithms to compute the optimal estimate iteratively - Jacobi, modified EPA and the Overlapping Subgraph Estimator (OSE) and discuss correctness and performance. Simulation studies are presented in section 4. The paper concludes with a summary in section 5.

2 The Optimal Estimate

Consider a measurement graph $\mathbf{G}$ with n nodes and m edges. Recall that k is the dimension of the node variables. Let $\mathbf{X}$ be a vector in $\mathbb{R}^{nk}$ obtained by stacking together all the node variables, known and unknown, i.e., $\mathbf{X} := [x_1^T, x_2^T, \ldots, x_n^T]^T$. Define $\mathbf{z} := [\zeta_1^T, \zeta_2^T, ..., \zeta_m^T]^T \in \mathbb{R}^{km}$ and $\boldsymbol{\epsilon} := [\epsilon_1^T, \epsilon_2^T, ..., \epsilon_m^T]^T \in \mathbb{R}^{km}$. This stacking together of variables allows us to rewrite (1) in the following form:

$$\mathbf{z} = \mathcal{A}^T \mathbf{X} + \boldsymbol{\epsilon}, \tag{2}$$

where $\mathcal{A}$ is a matrix uniquely determined by the graph. To construct $\mathcal{A}$, we start by defining the *incidence matrix A of the graph* $\mathbf{G}$, which is an $n \times m$ matrix with one row per node and one column per edge defined by $A := [a_{ue}]$, where a_{ue} is nonzero if and only if the edge $e \in \mathbf{E}$ is incident on the node $u \in \mathbf{V}$. When nonzero, $a_{ue} = -1$ if the edge e is directed towards u and $a_{ue} = 1$ otherwise. The matrix $\mathcal{A}$ that appears in (2) is an "expanded" version of the incidence matrix A, defined by $\mathcal{A} := A \otimes I_k$, where I_k is the $k \times k$ identity matrix and $\otimes$ denotes the Kronecker product.



$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \\ \zeta_6 \end{bmatrix} = \begin{bmatrix} I & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & -I & 0 \\ 0 & 0 & I & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & -I \\ 0 & 0 & 0 & I & 0 & -I \\ 0 & -I & 0 & 0 & 0 & I \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

Taking out the rows $1:k, k+1:2k$ and $4k+1:5k$ (corresponding to the reference nodes $1, 2, 5$), we construct $\mathcal{A}_r$; the remaining rows constitute $\mathcal{A}_b$. For this example, $\mathcal{A}_r^T \mathbf{x}_r = [x_1^T, -x_5^T, 0, x_5^T, 0, -x_2^T]^T$ and so eq. (3) becomes

$$\underbrace{\begin{bmatrix} \zeta_1 - x_1 \\ \zeta_2 + x_5 \\ \zeta_3 \\ \zeta_4 - x_5 \\ \zeta_5 \\ \zeta_6 + x_2 \end{bmatrix}}_{\bar{\mathbf{z}}} = \underbrace{\begin{bmatrix} -I & 0 & 0 \\ I & 0 & 0 \\ I & -I & 0 \\ 0 & 0 & -I \\ 0 & I & -I \\ 0 & 0 & I \end{bmatrix}}_{\mathcal{A}_b^T} \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ x_6 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}}_{\boldsymbol{\epsilon}}.$$

In the case when every measurement covariance is equal to the identity matrix, eq. (4) becomes

$$\begin{bmatrix} 3I & -I & 0 \\ -I & 2I & -I \\ 0 & -I & 3I \end{bmatrix} \begin{bmatrix} \hat{x}_3^* \\ \hat{x}_4^* \\ \hat{x}_6^* \end{bmatrix} = \begin{bmatrix} -\bar{\zeta}_1 + \bar{\zeta}_2 + \bar{\zeta}_3 \\ -\bar{\zeta}_3 + \bar{\zeta}_5 \\ -\bar{\zeta}_4 - \bar{\zeta}_5 + \bar{\zeta}_6 \end{bmatrix},$$

whose solution gives the optimal estimates of the unknown node variables x_3, x_4 and x_6.

Fig. 1. A measurement graph $\mathbf{G}$ with 6 nodes and 6 edges. Nodes $1, 2$ and 5 are reference nodes, which means that they know their own node variables.

By partitioning $\mathbf{X}$ into a vector $\mathbf{x}$ containing all the unknown node variables and another vector $\mathbf{x}_r$ containing all the known reference node variables: $\mathbf{X}^T = [\mathbf{x}_r^T, \mathbf{x}^T]^T$, we can re-write (2) as $\mathbf{z} = \mathcal{A}_r^T \mathbf{x}_r + \mathcal{A}_b^T \mathbf{x} + \boldsymbol{\epsilon}$, where $\mathcal{A}_r$ contains the rows of $\mathcal{A}$ corresponding to the reference nodes and $\mathcal{A}_b$ contains the rows of $\mathcal{A}$ corresponding to the unknown node variables. The equation above can be further rewritten as:

$$\bar{\mathbf{z}} = \mathcal{A}_b^T \mathbf{x} + \boldsymbol{\epsilon}, \tag{3}$$

where $\bar{\mathbf{z}} := \mathbf{z} - \mathcal{A}_r^T \mathbf{x}_r$ is a known vector. The optimal estimate (BLUE) $\hat{\mathbf{x}}^*$ of the vector of unknown node variables $\mathbf{x}$ for the measurement model 3 is the solution to the following system of linear equations:

$$\mathcal{L}\hat{\mathbf{x}}^* = \mathbf{b}, \tag{4}$$

where $\mathcal{L} := \mathcal{A}_b \mathscr{P}^{-1} \mathcal{A}_b^T$, $\mathbf{b} := \mathcal{A}_b \mathscr{P}^{-1} \bar{\mathbf{z}}$, and $\mathscr{P} := \mathrm{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]$ is the covariance matrix of the measurement error vector [1]. Since the measurement errors on two different edges are uncorrelated, $\mathscr{P}$ is a symmetric positive definite block diagonal matrix with the measurement error covariances along the diagonal: $\mathscr{P} = \mathrm{diag}(P_1, P_2, \ldots, P_m) \in \mathbb{R}^{km \times km}$, where $P_e = \mathrm{E}[\epsilon_e \epsilon_e^T]$ is the covariance of the measurement error ϵ_e.

The matrix $\mathcal{L}$ is invertible if and only if every weakly connected component of the graph $\mathbf{G}$ has at least one reference node [4]. A directed graph $\mathbf{G}$ is said to be weakly connected if there is a path from every node to every other node, not necessarily respecting the direction of the edges. In a weakly connected graph, the optimal estimate $\hat{\mathbf{x}}^*$ for every node u is unique for a given set of measurements $\mathbf{z}$. The error covariance of the optimal estimate $\boldsymbol{\Sigma} := \mathrm{E}[(\mathbf{x} - \hat{\mathbf{x}}^*)(\mathbf{x} - \hat{\mathbf{x}}^*)^T]$ is equal to $\mathcal{L}^{-1}$ and the $k \times k$ blocks on the diagonal of this matrix gives the estimation error covariances of the node variables.

Figure 1 shows an example of a measurement graph and the relevant equations.

3 Distributed Computation of the Optimal Estimate

In order to compute the optimal estimate $\hat{\mathbf{x}}^*$ by solving the equations (4) directly, one needs all the measurements and their covariances ($\mathbf{z}$, $\mathscr{P}$), and the topology of the graph ($\mathcal{A}_b, \mathcal{A}_r$). In this section we consider iterative distributed algorithms to compute the optimal estimates for the measurement model (2). These algorithms compute the optimal estimate through multiple iterations, with the constraint that a node is allowed to communicate only with its neighbors. The concept of "neighbor" is determined by the graph $\mathbf{G}$, in the sense that two nodes are *neighbors* if there is an edge in $\mathbf{G}$ between them (in either direction). This implicitly assumes bidirectional communication. We describe three algorithms - Jacobi, modified EPA, and OSE, the last being the novel contribution of this paper. We will see that OSE algorithm converges even when communication faults destroy the bidirectionality of communication.

3.1 The Jacobi Algorithm

Consider a node u with unknown node variable x_u and imagine for a moment that the node variables for all neighbors of u are exactly known and available to u. In this case, u could compute its optimal estimate by simply using the measurements between itself and its 1-hop neighbors. This estimation problem is fundamentally no different than the original problem, except that it is defined over the much smaller graph $\mathbf{G}_u(1) = (\mathbf{V}_u(1), \mathbf{E}_u(1))$, whose node set $\mathbf{V}_u(1)$ include u and its 1-hops neighbors and the edge set $\mathbf{E}_u(1)$ consists of only the edges between u and its 1-hops neighbors. We call $\mathbf{G}_u(1)$ the *1-hop subgraph of* $\mathbf{G}$ *centered at* u. Since we are assuming that the node variables of all neighbors of u are exactly known, all these nodes should be understood as references.

In the Jacobi algorithm, at every iteration, a node gathers the estimates of its neighbors from them by exchanging messages and updates it own estimate by solving the optimal estimation problem in the 1-hop subgraph $\mathbf{G}_u(1)$ by taking the estimates of its neighbors as the true values (reference variables). It turns out that this algorithm corresponds exactly to the Jacobi algorithm for the iterative solution to the linear equation (4) and is guaranteed to converge to the true solution of (4) when the iteration is done in a synchronous manner [5]. When done asynchronously, or in the presence of communication failures, it is guaranteed to converge under additional mild assumptions [5]. The algorithm can be terminated at a node when the change in its recent estimate is seen to be lower than a certain pre-specified threshold value, or when a certain maximum number of iterations are completed. The details of the Jacobi algorithm can be found in [2, 5].

Note that to compute the update $\hat{x}_u^{(i+1)}$, node u also needs the measurements and associated covariances ζ_e, P_e on the edges $e \in \mathbf{E}_u(1)$ of its 1-hop subgraph. We assume that after deployment of the network, nodes detect their neighbors and exchange their relative measurements as well as the associated covariances. Each node uses this information obtained initially for all future computation.

3.2 Modified EPA

The Embedded Polygon Algorithm (EPA) proposed in [8] can be used for iteratively solving (4); since it is essentially a block – Jacobi method of solving a system of linear equations, where the blocks correspond to non-overlapping polygons. The special case when the polygons are triangles has been extensively studied in [8]. We will not include here the details of the algorithm, including triangle formation in the initial phase, the intermediate computation, communication and update. The interested reader is referred to [8]. It is not difficult to adapt the algorithm in [8] to the problem considered in this paper. We have implemented the modified EPA algorithm (with triangles as the embedded polygons) and compared it with both Jacobi and OSE. Results are presented in section 4.

3.3 The Overlapping Subgraph Estimator Algorithm

The Overlapping Subgraph Estimator (OSE) algorithm achieves faster convergence than Jacobi and modified EPA, while retaining their scalability and robustness properties. The OSE algorithm is inspired by the multisplitting and Weighted Additive Schwarz method of solving linear equations [13].

The OSE algorithm can be thought of as an extension of the Jacobi algorithm, in which individual nodes utilize larger subgraphs to improve their estimates. To understand how this can be done, suppose that each node broadcasts to its neighbors not only is current estimate, but also all the latest estimates that it received from his neighbors. In practice, we have a simple two-hop communication scheme and, in the absence of drops, at the ith iteration step, each node will have the estimates $\hat{x}_v^{(i)}$ for its 1-hop neighbors and the (older) estimates $\hat{x}_v^{(i-1)}$ for its 2-hop neighbors (i.e., the nodes at a graphical distance of two).

Under this information exchange scheme, at the ith iteration, each node u has estimates of all node variables in the set $\mathbf{V}_u(2)$ consisting of itself and all its 1-hop and 2-hop neighbors. In the OSE algorithm, each node updates its estimate using the *2-hop subgraph centered at* u $\mathbf{G}_u(2) = (\mathbf{V}_u(2), \mathbf{E}_u(2))$, with edge set $\mathbf{E}_u(2)$ consisting all the edges of the original graph $\mathbf{G}$ that connect element of $\mathbf{V}_u(2)$. For this estimation problem, node u takes as references the variables of the nodes at the "boundary" of its 2-hop subgraph: $\mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$. These nodes are at a graphical distance of 2 from u. We assume that the nodes use the first few rounds of communication to determine and communicate to one another the measurements and associated covariances of their 2-hop subgraphs. The *OSE algorithm* can be summarized as follows:

1. Each node $u \in \mathbf{V}$ picks arbitrary initial estimates $\hat{x}_v^{(-1)}$, $v \in \mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$ for the node variables of all its 2-hop neighbors. These estimates do not necessarily have to be consistent across the different nodes.
2. At the ith iteration, each node $u \in \mathbf{V}$ assumes that the estimates $\hat{x}_v^{(i-2)}$, $v \in \mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$ (that it received through its 1-hop neighbors) are correct and solves the corresponding optimal estimation problem associated with the 2-hop subgraph $\mathbf{G}_u(2)$. In particular, it solves the following linear equations: $\mathcal{L}_{u,2}\mathbf{y}_u = \mathbf{b}_u$, where $\mathbf{y}_u$ is a vector of node variables that correspond to the nodes in its 1-hop subgraph $\mathbf{G}_u(1)$, and $\mathcal{L}_{u,2}, \mathbf{b}_u$ are defined for the subgraph $G_u(2)$ as $\mathcal{L}, \mathbf{b}$ were for $\mathbf{G}$ in eq. (4). After this computation, node u updates its estimate as $\hat{x}_u^{(i+1)} \leftarrow \lambda y_u + (1-\lambda)\hat{x}_u^{(i)}$, where $0 < \lambda \leq 1$ is a pre-specified design parameter and y_u is the variable in $\mathbf{y}_u$ that corresponds to x_u. The new estimate $\hat{x}_u^{(i+1)}$ as well as the estimates $\hat{x}_v^{(i)}$, $v \in \mathbf{V}_u(1)$ previously received from its 1-hop neighbors are then broadcasted by u to all its 1-hop neighbors.
3. Each node then listens for the broadcasts from its neighbors, and uses them to update its estimates for the node variables of all its 1-hop and 2-hop neighbors $\mathbf{V}_u(2)$. Once all updates are received a new iteration can start.

The termination criteria will vary depending on the application, as discussed for the Jacobi algorithm. As in the case of Jacobi, we assume that nodes exchange

measurement and covariance information with their neighbors in the beginning, and once obtained, uses those measurements for all future time.

As an illustrative example of how the OSE algorithm proceeds in practice, consider the measurement graph shown in figure 2(a) with node 1 as the single reference. Figure 2(b) shows the 2-hop subgraph centered at node 4, $\mathbf{G}_4(2)$, which consists of the following nodes and edges: $\mathbf{V}_4(2) = \{1, 3, 5, 4, 6, 2\}$ and $\mathbf{E}_4(2) = \{1, 2, 3, 4, 5, 6\}$. Its 2-hop neighbors are $\mathbf{V}_4(2) \setminus \mathbf{V}_4(1) = \{1, 2, 5\}$. After the first round of inter node communication, node 4 has the estimates of its neighbors 3 and 6: $x_3^{(0)}, x_6^{(0)}$ (as well as the measurements ζ_3, ζ_5 and covariances P_3, P_5). After the second round of communication, node 4 has the node estimates $x_1, \hat{x}_3^{(1)}, \hat{x}_5^{(0)}, \hat{x}_6^{(1)}, \hat{x}_2^{(0)}$ (and the measurements $\zeta_1, \ldots, \zeta_6$ and covariances $P_1, \ldots, P_6$). Assuming no communication failures, at every iteration i, node 4 uses $x_1, \hat{x}_3^{(i-2)}$ and $\hat{x}_5^{(i-2)}$ as the reference variables and computes "temporary" estimates y_3, y_4, y_6 (of x_3, x_4 and x_6) by solving the optimal estimation problem in its 2-hop subgraph. It updates its estimate as : $\hat{x}_4^{(i+1)} \leftarrow \lambda y_4 + (1 - \lambda)\hat{x}_4^{(i)}$, and discards the other variables computed.

Note that all the data required for the computation at a node is obtained by communicating with its 1-hop neighbors. Convergence to the optimal estimate will be discussed in section 3.5.

Remark 1 (h-hop OSE algorithm). One could also design a *h-hop OSE* algorithm by letting every node utilize a h-hop subgraph centered at itself, where h is some (not very large) integer. This would be a straightforward extension of the 2-hop OSE just described, except that at every iteration, individual nodes would have to transmit larger amounts of data than in 2-hop OSE, potentially requiring multiple packet transmissions at each iteration. In practice, this added communication cost will limit the allowable value of h. □

The Jacobi, EPA and OSE algorithms are all iterative methods to compute the solution of a system of linear equations. The Jacobi and EPA are similar in nature, EPA essentially being a block-Jacobi method. The OSE is built upon the Filtered Schwarz method [2], which is a refinement of the Schwarz method [13]. The OSE algorithm's gain in convergence speed with respect to the Jacobi and modified EPA algorithms comes from the fact that the 2-hop subgraphs $\mathbf{G}_u(2)$ contain more edges than the 1-hop subgraphs $\mathbf{G}_u(1)$, and the subgraphs of different nodes are overlapping. It has been observed that a certain degree of overlap may lead to a speeding up of the Schwarz method [13].

Improving Performance Through Flagged Initialization. One can further improve the performance of OSE (and also of Jacobi and modified EPA) by providing a better initial condition to it, which does not require more communication or computation. After deployment of the network, the reference nodes initialize their variables to their known values and every other node initializes its estimate to ∞, which serves as a flag to declare that it has no estimate. In the subsequent updates of a node's variables, the node only includes in its 1- or 2-hop subgraph those nodes that have finite estimates. If none of the neighbors have a

finite estimate, then a node keeps its estimate at ∞. In the beginning, only the references have finite estimates. In the first iteration, only the neighbors of the references compute finite estimates by looking at their 1-hop subgraphs. In the next iteration, their neighbors do the same by looking at their 2 hop subgraphs and so on until all nodes in the graph have finite estimates. In general, the time it takes for all nodes to have finite estimates will depend on the radius of the network (the minimum graphical distance between any node and the nearest reference node). Since the flagged initialization only affects the initial stage of the algorithms, it does not affect their convergence properties.

3.4 Asynchronous Updates and Link Failures

In the previous description of the OSE algorithm, we assumed that communication was essentially synchronous and that all nodes always received broadcasts from all their neighbors. However, the OSE algorithm also works with *link failures* and *lack of synchronization* among nodes. To achieve this a node broadcasts its most recent estimate and waits for a certain time-out period to receive data from its neighbors. It proceeds with the estimate update after that period, even if it does not receive data from all of its neighbors, by using the most recent estimates that it received from its neighbors. A node may also receive multiple estimates of another node's variable. In that case, it uses the most recent estimate, which can be deduced by the time stamps on the messages. The Jacobi and the modified EPA algorithms can similarly be made robust to link failures [5, 8].

In practice nodes and communication links may fail temporarily or permanently. A node may simply remove from its subgraph those nodes and edges that have failed permanently (assuming it can be detected) and carry on the estimation updates on the new subgraph. However, if a node or link fails permanently, the measurement graph changes permanently, requiring redefinition of the optimal estimator. To avoid this technical difficulty, in this paper we only consider temporary link failures, which encompasses temporary node failures.

3.5 Correctness

An iterative algorithm is said to be *correct* if the estimate produced by the algorithm $\mathbf{x}^{(i)}$ converges to the true solution $\hat{\mathbf{x}}^*$ as the number of iterations i increase, i.e., $\|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^*\| \to 0$ as $i \to \infty$. The assumption below is needed to prove correctness of the OSE algorithm, which is stated in Theorem 1.

Assumption 1. *At least one of the statements below holds:*

1. *All covariance matrices P_e, $e \in \mathbf{E}$ are diagonal.*
2. *All covariance matrices P_e, $e \in \mathbf{E}$ are equal.* □

Theorem 1 (Correctness of OSE). *When assumption 1 holds, the OSE algorithm converges to the optimal estimate $\hat{\mathbf{x}}^*$ as long as there is a number ℓ_d such that the number of consecutive failures of any link is less than ℓ_d.* □

We refer the interested reader to [2] for a proof of this result.

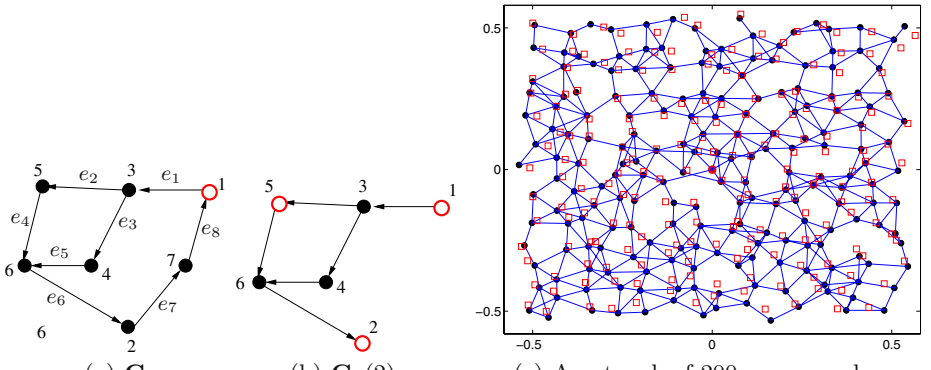
Remark 2. Assumption 1 is too restrictive in certain cases. In particular, the simulations reported in Section 4 were done with covariance matrices that did not satisfy this assumption, yet the algorithm was seen to converge in all the simulations. The reason is that this assumption is needed so that sufficient conditions for convergence are satisfied [2], and is not necessary in general.

3.6 Performance

Since minimizing energy consumption is critically important in sensor networks, we choose as performance metric of the algorithms the total energy consumed by a node before a given normalized error is achieved. The *normalized error* $\varepsilon^{(i)}$ is defined as

$$\varepsilon^{(i)} := \|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^*\| / \|\hat{\mathbf{x}}^*\|$$

and is a measure of how close the iterate $\hat{\mathbf{x}}^{(i)}$ is to the correct solution $\hat{\mathbf{x}}^*$ at the end of iteration i. We assume that nodes use broadcast communication to send data.



(a) $\mathbf{G}$ (b) $\mathbf{G}_4(2)$ (c) A network of 200 sensor nodes.

Fig. 2. (a) A measurement graph $\mathbf{G}$ with node 1 as reference, and (b) a 2-hop subgraph $\mathbf{G}_4(2)$ centered at node 4. While running the OSE algorithm, node 4 treats 1, 5 and 2 as reference nodes in the subgraph $\mathbf{G}_4(2)$ and solves for the unknowns x_3, x_4 and x_6. (c) A sensor network with 200 nodes in a unit square area. The edges of the measurement graph are shown as line segments connecting the true nodes positions, which are shown as black circles. Two nodes with an edge between them have a noisy measurement of their relative positions in the plane. The little squares are the positions estimated by the (centralized) optimal estimator. A single reference node is placed at $(0, 0)$.

As discussed in section 1.1, we take the number of packets transmitted and received by a node as a measure of energy consumption. Let $N_{tx}^{(i)}(u)$ be the number of packets a node u transmits to its neighbors during the ith iteration.

The energy $E^{(i)}(u)$ expended by u in sending and receiving data during the ith iteration is computed by the following formula:

$$E^{(i)}(u) = N_{tx}^{(i)}(u) + \frac{3}{4} \sum_{v \in \mathcal{N}_u} N_{tx}^{(i)}(v), \tag{5}$$

where N_u is the set of neighbors of u. The factor 3/4 is chosen to account for the ratio between the power consumptions in the receive mode and the transmit mode. Our choice is based on values reported in [10] and [14]. The average energy consumption $\bar{E}(\epsilon)$ is the average (over nodes) of the total of energy consumed among all the nodes till the normalized error reduces to ϵ. For simplicity, eq. (5) assumes synchronous updates and perfect communication (no retransmissions). When packet transmission is unsuccessful, multiple retransmissions maybe result, making the resulting energy consumption a complex function of the parameters involved [11, 10].

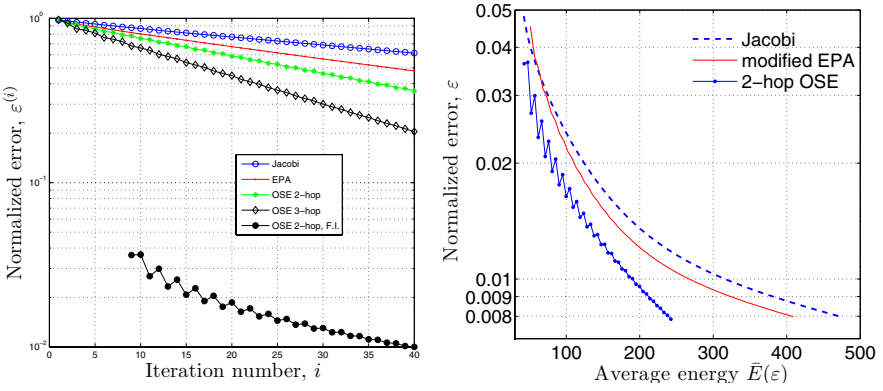
In one iteration of the Jacobi algorithm, a node needs to broadcast its own estimate, which consists of k real numbers. Recall that k is the dimension of the node variables. Assuming a 32 bit encoding, that amounts to $4k$ bytes of data. In the OSE algorithm, a node with d neighbors has to broadcast data consisting of $4d$ bytes for its neighbors' IP addresses, $4k(d+1)$ bytes for the previous estimates of itself and its neighbors, and $3d$ bytes for time stamps of those estimates. This leads to a total of $(7+4k)d+4k$ bytes of data, and consequently the number of packets in a message becomes

$$N_{tx}(u) = \lceil \frac{(7+4k)d+4k}{max\_databytes\_pkt} \rceil, \tag{6}$$

where $max\_databytes\_pkt$ is the maximum number of bytes of data allowed in the payload per packet. In this paper we assume that the maximum data per packet is 118 bytes, as per IEEE 802.15.4 specifications [15]. For comparison, we note that the number of bytes in a packet transmitted by MICA motes can vary from 29 bytes to 250 bytes depending on whether B-MAC or S-MAC is used [16]. If the number of data bytes allowed is quite small, OSE may require multiple packet transmission in every iterations, making it more expensive.

4 Simulations

For simulations reported in this section, we consider location estimation as an application of the problem described in this paper. The node variable x_u is node u's position in 2-d Euclidean space. We present a case study with a network with 200 nodes that were randomly placed in an area approximately 1×1 area (Figure 2(c)). Some pairs of nodes u, v that were within a range of less than $r_{\max} = 0.11$ were allowed to have measurements of each others' relative distance r_{uv} and bearing θ_{uv}. Node 1, placed at $(0, 0)$ was the only reference node. Details of the noise corrupting the measurements and the resulting covariances can be found in [2]. The locations estimated by the (centralized) optimal estimator are shown in Figure 2(c) together with the true locations.



(a) Normalized error vs. iteration number.

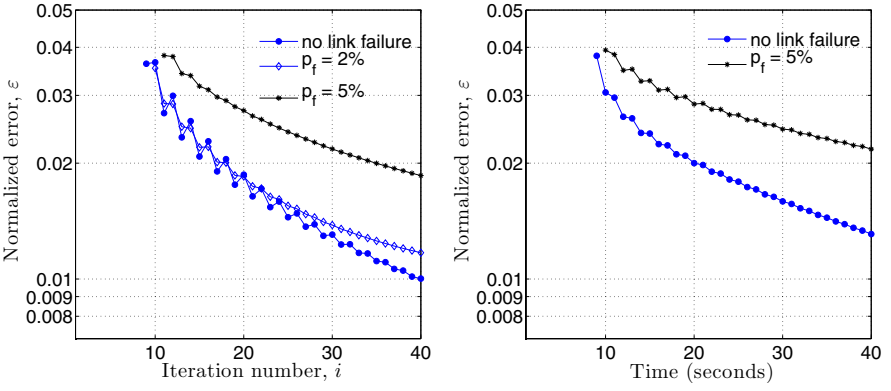
(b) Normalized error vs. average energy consumed.

Fig. 3. Performance comparison of the three algorithms. (a) shows the reduction in normalized error with iteration number for the three algorithms, and also the drastic reduction in the error with flagged initialization for the 2-hop OSE (the legend "F.I." refers to flagged initialization). The simulations without flagged initialization were done with all initial node position estimates set to $(0, 0)$. (b) shows the Normalized error vs. average energy consumption of 2-hop OSE, modified EPA and Jacobi with broadcast communication. Flagged initialization was used in all the three algorithms. All simulations shown in (a) and (b) were done in Matlab.

Simulations were done both in Matlab and in the network simulator package GTNetS [17]. The Matlab simulations were done in a synchronous manner. The purpose of the synchronous simulations was to compare the performance of the three algorithms – Jacobi, modified EPA and OSE – under exactly the same conditions. Synchronous Matlab simulations with link failure were conducted to study the effect of communication faults (in isolation from the effect of asynchronism). The GTNetS simulations were done to study OSE's performance in a more realistic setting, with asynchronous updates and faulty communication. For all OSE simulations, λ was chosen (somewhat arbitrarily) as 0.9.

Figure 3(a) compares the normalized error as a function of iteration number for the three algorithms discussed in this paper - Jacobi, EPA and the OSE. Two versions of OSE were tested, 2-hop and 3-hop. It is clear from this figure that the OSE outperforms both Jacobi and modified EPA. As the figure shows, drastic improvement was achieved with the flagged initialization scheme. With it, the 2-hop OSE was able to estimate the node positions within 3% of the optimal estimate after 9 iterations. For the flagged OSE, the normalized error is not defined till iteration number 8, since some nodes had no estimate of their positions till that time.

The Performance of the three algorithms - Jacobi, modified EPA and 2-hop OSE are compared in terms of the average energy consumption $\bar{E}$ in Figure 3(b). Flagged initialization was used in all three algorithms. To compute the energy consumption



(a) Synchronous simulation (b) Async. GTNetS simulation

Fig. 4. (a)Normalized error as a function of iteration number in the presence of link failures. Two different failure probabilities are compared with the case of no failure. (b) Normalized error vs. Time (seconds) for asynchronous 2-hop OSE simulations conducted in GTNetS, with and without link failure. As expected, performance in the asynchronous case is slightly poorer than in the corresponding synchronous case.

for the 2-hop OSE, we apply (6) with $k = 2$ and $max\_databytes\_pkt = 118$ to get $N_{tx}(u) = \lceil (15d_u + 8)/118 \rceil$. The average node degree being 5, the number of packets broadcasted per iteration in case of the OSE algorithm was 1 for almost all the nodes. For Jacobi, the number of packets broadcasted at every iteration was 1 for every node. For the modified EPA algorithm, the number of packets in every transmission was 1 but the total number of transmissions in every iteration were larger (than Jacobi and OSE) due to the data exchange required in both the EPA update and EPA solve steps (see [8] for details). The normalized error against the average (among all the nodes) total energy consumed $\bar{E}$ is computed and plotted in Figure 3(b). Comparing the plots one sees that for a normalized error of 1%, the OSE consumes about 70% of the energy consumed by modified EPA and 60% of that by Jacobi. For slightly lower error, the difference is more drastic: to achieve a normalized error of 0.8%, OSE needs only 60% of the energy consumed by EPA and about half of that by Jacobi.

Note that the energy consumption benefits of OSE become more pronounced as one asks for higher accuracy, but less so for low accuracy. This is due to flagged initialization, which accounts for almost all the error reduction in the first few iterations.

To simulate faulty communication, we let every link fail independently with a probability p_f that is constant for all links during every iteration. Figure 4(a) shows the normalized error as a function of iteration number (from three representative runs) for two different failure-probabilities: $p_f = 0.025$ and 0.05. In all the cases, flagged initialization was used. The error trends show the algorithm converging with link failures. As expected, though, higher failure rates resulted in deteriorating performance.

The OSE algorithm was also implemented in the GTNetS simulator [17], and the results for the 200 node network are shown in Figure 4(b). Each node sleeps until it receives the first packet from a neighbor, after which it updates its estimate and sends data to its neighbors every second. Estimates are updated in an asynchronous manner, without waiting to receive data from all neighbors. Time history of the normalized error is shown in Figure 4(b). Both failure-free and faulty communication (with $p_f = 0.05$) cases were simulated. Even with realistic asynchronous updates and link failures, the OSE algorithm converges to the optimal estimate. Since the nodes updated their estimates every second, the number of seconds (x-axis in Figure 4(b)) can be taken approximately as the number of iterations. Comparing Figure 4(a) and (b), we see that the convergence in the asynchronous case is slightly slower than in the synchronous case.

5 Conclusions

We have developed a distributed algorithm that iteratively computes the optimal estimate of vector valued node variables, when noisy difference of variables between certain pairs of nodes are available as measurements. This situation covers a range of problems relevant to sensor network applications, such as localization and time synchronization. The optimal estimate produces the minimum variance estimate of the node variables from the noisy measurements among all linear unbiased estimates. The proposed Overlapping Subgraph Estimator (OSE) algorithm computes the optimal estimate iteratively. The OSE algorithm is distributed, asynchronous, robust to link failures and scalable. The performance of the algorithm was compared to two other iterative algorithms – Jacobi and modified EPA. The OSE outperformed both of these algorithms, consuming much less energy for the same normalized error. Simulations with a simple energy model indicate that OSE can potentially cut down energy consumption by a factor of two or more compared to Jacobi and modified EPA.

There are many avenues of future research. Extending the algorithm to handle correlated measurements and developing a distributed algorithm for computing the covariance of the estimates are two challenging tasks that we leave for future work.

Bibliography

[1] Mendel, J.M.: Lessons in Estimation Theory for Signal Processing, Communications and Control. Prentice Hall P T R (1995)

[2] Barooah, P., da Silva, N.M., Hespanha, J.P.: Distributed optimal estimation from relative measurements: Applications to localizationa and time synchronization. Technical report, Univ. of California, Santa Barbara (2006)

[3] Karp, R., Elson, J., Estrin, D., Shenker, S.: Optimal and global time synchronization in sensornets. Technical report, Center for Embedded Networked Sensing,Univ. of California, Los Angeles (2003)

[4] Barooah, P., Hespanha, J.P.: Optimal estimation from relative measurements: Electrical analogy and error bounds. Technical report, University of California, Santa Barbara (2003)
[5] Barooah, P., Hespanha, J.P.: Distributed optimal estimation from relative measurements. In: 3rd ICISIP, Bangalore, India (2005)
[6] Chintalapudi, K., Dhariwal, A., Govindan, R., Sukhatme, G.: Ad-hoc localization using ranging and sectoring. In: IEEE Infocom. (2004)
[7] Moore, D., Leonard, J., Rus, D., Teller, S.: Robust distributed network localization with noisy range measurements. In: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems. (2004)
[8] Delouille, V., Neelamani, R., Baraniuk, R.: Robust distributed estimation in sensor networks using the embedded polygon algorithms. In: IPSN. (2004)
[9] Min, R., Bhardwaj, M., Cho, S., Sinha, A., Shih, E., Sinha, A., Wang, A., Chandrakasan, A.: Low-power wireless sensor networks. In: Keynote Paper ESSCIRC, Florence, Italy (2002)
[10] Bougard, B., Catthoor, F., Daly, D.C., Chandrakasan, A., Dehaene, W.: Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In: Design, Automation and Test in Europe (DATE). (2005)
[11] Carvalho, M.M., Margi, C.B., Obraczka, K., Garcia-Luna-Aceves, J.: Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks. In: IEEE ICCCN. (2004)
[12] Shih, E., Cho, S., Fred S. Lee, B.H.C., Chandrakasan, A.: Design considerations for energy-efficient radios in wireless microsensor networks. Journal of VLSI Signal Processing **37** (2004) 77–94
[13] Frommer, A., Schwandt, H., Szyld, D.B.: Asynchronous weighted additive Schwarz methods. Electronic Transactions on Numerical Analysis **5** (1997) 48–61
[14] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: Proceedings of the IEEE Infocom. (2002)
[15] IEEE 802.15 TG4: IEEE 802.15.4 specifications (2003) `http://www.ieee802.org/15/pub/TG4.html`.
[16] Ault, A., Zhong, X., Coyle, E.J.: K-nearest-neighbor analysis of received signal strength distance estimation across environments. In: 1st workshop on Wireless Network Measurements, Riva Del Garda, Italy (2005)
[17] Riley, G.F.: The Georgia Tech Network Simulator. In: Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools). (2003)