DISTRIBUTED ESTIMATION WITH AD HOC WIRELESS SENSOR NETWORKS†

Ioannis D. Schizas, Alejandro Ribeiro, and Georgios B. Giannakis

Department of Electrical and Computer Engineering, University of Minnesota
200 Union Street SE, Minneapolis, Minnesota, USA 55455.
Phone: + (001) 612-626-7781, Fax: + (001) 612-625-4583
Email: schiz001@umn.edu, aribeiro@ece.umn.edu, georgios@ece.umn.edu

ABSTRACT

We consider distributed estimation of a deterministic parameter vector using an ad hoc wireless sensor network. The estimators derived are obtained as solutions of constrained convex optimization problems. Using the method of multipliers in conjunction with a block coordinate descent approach we demonstrate how the resultant algorithms can be decomposed into a set of simpler tasks suitable for distributed implementation. We show that these algorithms have guaranteed convergence to properly defined optimum estimators, and further exemplify their applicability to solving estimation problems where the signal model is completely or partially known at individual sensors. Through numerical experiments we illustrate that our algorithms outperform existing alternatives.

1. INTRODUCTION

A major challenge in wireless sensor networks (WSNs) is the computation of parameter estimates based on distributed observations collected at individual sensors. Severe energy and bandwidth limitations call for the design and implementation of distributed algorithms that are efficient in terms of reducing communication overhead and computational cost.

A recently introduced class of distributed estimation algorithms is based on successive refinement of local estimates maintained at individual sensors. These approaches rely on communication with one-hop neighbors only, to develop iterative algorithms that eventually converge to the desired estimate. In a nutshell, each iteration comprises a transmission step in which sensors communicate certain information to their neighbors, and an update step in which the information collected from all neighbors is used to update the local estimate. The notion of *consensus averaging* for the estimation of deterministic unknown parameters using linear data models was introduced in [1,7,8], whereby each sensor updates its local estimate by appropriate weighting the estimates of its neighbors. A more elaborate approach entailing distributed computation of the sample average estimator with the aid of dual decomposition techniques was studied in [4]. For distributed estimation of a Gaussian random parameter in a scalar linear model, [2] applied the Jacobi iteration. The same scalar linear model in a dynamical system was also considered in [6]. While different in focus, these works share the common thread of being successive refinement algorithms based on communicating information with one-hop neighbors only.

In the present paper, we deal with estimation of unknown deterministic parameters of linear (but not necessarily) Gaussian observation models. We further consider cases where the signal model is completely or partially known at individual sensors. Our approach amounts to recasting estimators of interest as minimizers of convex functions under a set of linear constraints. Quite surprisingly, we are able to formulate these convex optimization problems in a form that is amenable to parallel – i.e., distributed – computation. We first consider estimation when the observation model is known at each sensor; in some sense generalizing the work of [8] on distributed consensus (Section 3). To solve this problem we utilize the method of multipliers to find the optimal solution as the minimum of the augmented Lagrangian function. We then use a block coordinate descent algorithm to decompose the minimization of the augmented Lagrangian into simple separable tasks [1] leading to a distributed successive refinement algorithm (Section 3.1). Subsequently, we consider the sensors having partial knowledge of the signal model (Section 4). While inherently more complicated, the same steps of i) recasting the estimator as the minimum of a convex function; ii) applying the method of multipliers; and iii) separating the problem with a block coordinate descent algorithm, lead to a distributed successive refinement algorithm with guaranteed convergence to the optimal estimate. We provide corroborating simulations in Section 5 and conclude the paper in Section 6.

2. PROBLEM FORMULATION

Consider an ad-hoc WSN with M sensors. Communication links in the WSN are represented by a graph whose vertices are the sensors and its edges are formed by the available communication links; see Fig. 1. The set of sensors having an active link with the j-th sensor comprise the neighborhood $\mathcal{N}_j$. The WSN is deployed to estimate a $p\times 1$ parameter vector $\mathbf{s}$ based on distributed observations $\mathbf{x}_j \in \mathbb{R}^{L_j\times 1}$ with $\mathbf{x}_j$ taken at the j-th sensor. Observations are related to the unknown vector by the linear model

$$\mathbf{x}_j = \mathbf{H}_j\mathbf{s}+\mathbf{n}_j, \quad j\in[1,M] \tag{1}$$

where $\mathbf{H}_j \in \mathbb{R}^{L_j\times p}$, and the zero-mean noise $\mathbf{n}_j \in \mathbb{R}^{L_j\times 1}$ has covariance $\mathbf{\Sigma}_{n_jn_j} := \mathrm{E}[\mathbf{n}_j\mathbf{n}_j^T]$. Defining $\mathbf{x} := [\mathbf{x}_1^T\ldots\mathbf{x}_M^T]^T$, $\mathbf{H} = [\mathbf{H}_1^T,\ldots,\mathbf{H}_M^T]^T$ and $\mathbf{\Sigma}_{nn} = E[\mathbf{n}\mathbf{n}^T]$, the minimum variance (best) linear unbiased estimator (BLUE) is [3]

$$\hat{\mathbf{s}} := \left(\mathbf{H}^T\mathbf{\Sigma}_{nn}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{\Sigma}_{nn}^{-1}\mathbf{x} := \mathbf{C}\mathbf{x}, \tag{2}$$

where $\mathbf{C} := \left(\mathbf{H}^T\mathbf{\Sigma}_{nn}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{\Sigma}_{nn}^{-1}$ and $\mathbf{H}$ is assumed to be full rank. Linear estimators are particularly attractive given the limited computing capabilities of the sensors. If the $\{\mathbf{n}_j\}_{j=1}^M$ are normally distributed, $\hat{\mathbf{s}}$ in (2) is also the minimum variance unbiased estimator (MVUE) among all (not necessarily linear) estimators.

Notice that the information contained in $\mathbf{x}$ is scattered around the sensor network. One approach to computing the desired $\hat{\mathbf{s}}$ is to transmit $\mathbf{x}_j$ for $j=1,\ldots,M$ to a fusion center (FC) and then directly compute $\hat{\mathbf{s}}$ using (2). Besides being communication costly, this approach is also prone to FC failures. In this work, we will develop decentralized algorithms for solving (2) with different degrees of knowledge about $\mathbf{C}$. Specifically, we consider two scenarios:

(s1) With $\mathbf{C} := [\mathbf{C}_1,\ldots,\mathbf{C}_M]$, we assume that sensor j has available its corresponding part $\mathbf{C}_j$. This may be the case when $\mathbf{H}$ and $\mathbf{\Sigma}_{nn}$ are known to every sensor. Moreover, if $\mathbf{H}_j = \mathbf{I}_p$

† Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

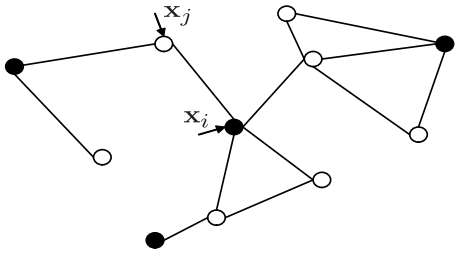

Figure 1: An ad-hoc wireless sensor network.

is the $p \times p$ identity matrix and $\Sigma_{n_j n_j} = \sigma^2 \mathbf{I}_p$, we have that $\mathbf{C}_j = \mathbf{I}_p/M$. The latter is assumed by the vector consensus averaging setup in [8].

(s2) The j-th sensor has available its corresponding part of the signal model; i.e., $\mathbf{H}_j$ and $\Sigma_{n_j n_j}$. This problem is considered in [7]. Matrices $\Sigma_{n_j n_j}$ can be found in sensor j by sample averaging, while $\mathbf{H}_j$ can be obtained via model estimation.

Our approach is to write the estimators as the solution of an appropriate optimization problem, and then use convex optimization techniques to split the original problem into simpler subtasks that can be implemented in parallel [1]. As in e.g., [2,7,8], we assume that: i) the communication channels between neighbor sensors are ideal; and ii) the communication graph of the WSN is connected.

3. GENERALIZED CONSENSUS AVERAGE

We begin by considering scenario (s1) whereby we want to compute the BLUE estimator in (2) when $\mathbf{C}_j$ is available at sensor j. It follows easily that the quantity $\hat{\mathbf{s}} = \sum_{j=1}^M \mathbf{C}_j \mathbf{x}_j$ is the solution of the following minimization problem:

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \mathbb{R}^{p \times 1}} \sum_{j=1}^M \|\mathbf{s} - M\mathbf{C}_j \mathbf{x}_j\|_2^2. \tag{3}$$

Notice that (3) is formulated in terms of the variable $\mathbf{s}$ making it an unlikely candidate for distributed implementation. This prompt us to define a subset of nodes $\mathscr{A} \subseteq [1,M]$ and modify (3) as

$$\begin{aligned} \{\hat{\mathbf{s}}_j\}_{j=1}^M := \min & \sum_{j=1}^M \|\mathbf{s}_j - M\mathbf{C}_j \mathbf{x}_j\|_2^2 \\ \text{s.t. } & \mathbf{s}_j = \bar{\mathbf{s}}_l,\ l \in \mathscr{A},\ j \in \mathscr{N}_l \end{aligned} \tag{4}$$

where we associate the variable $\mathbf{s}_j$ with the j-th sensor. We can interpret $\hat{\mathbf{s}}_j$ as the local estimate and ideally we would like to have $\hat{\mathbf{s}}_j = \hat{\mathbf{s}}$ for all j.

The latter would be the case if we impose the constraint $\mathbf{s}_1 = \ldots = \mathbf{s}_M$ which amounts to setting $\mathscr{A} = [1,M]$ in (4). Indeed, if $\mathbf{s}_1 = \ldots = \mathbf{s}_M$ then the arguments in (3) and (4) coincide and so do the optima. It turns out, that a relaxed requirement on $\mathscr{A}$ suffices to guarantee the equivalence of (3) and (4). This requirement is introduced in the following definition.

Definition 1 *We say that $\mathscr{A}$ is a bridge sensor set if and only if,*

(a) *$\forall\ j \in [1,M]$ there exists at least one $l \in \mathscr{A}$ so that $l \in \mathscr{N}_j$; and*

(b) *for every sensor $l_1 \in \mathscr{A}$ there exists a sensor $l_2 \in \mathscr{A}$ such that the shortest path between l_1 and l_2 has at most two edges.*

As an example, consider the WSN in Fig. 1 where the black sensors represent a possible selection of the set $\mathscr{A}$ to be a bridge set of sensors. The set of bridge neighbors j will be denoted as $\mathscr{M}_j := \mathscr{N}_j \cap \mathscr{A}$.

The equivalence between (3) and (4) when $\mathscr{A}$ is a bridge set of sensors is claimed in the following proposition.

Proposition 1 *If $\mathscr{A}$ is a bridge set of sensors, the optimization problems* (3) *and* (4) *are equivalent in the sense that*

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}_j,\ \forall j \in [1,M] \tag{5}$$

with $\hat{\mathbf{s}}$ the solution of (3) *and $\{\hat{\mathbf{s}}_j\}_{j=1}^M$ the solution of* (4).

Proof: We will show first that the constraints $\mathbf{s}_j = \bar{\mathbf{s}}_l$ for $l \in \mathscr{A}$ and $j \in \mathscr{N}_l$ are equivalent to $\mathbf{s}_{j_1} = \mathbf{s}_{j_2}$. To this end, consider $l_1, l_2 \in \mathscr{A}$ with $l_1 \in \mathscr{N}_{j_1}$ and $l_2 \in \mathscr{N}_{j_2}$, with the existence of l_1, l_2 guaranteed by $\mathscr{A}$'s definition [c.f. Def. 1-(a)]. From the constraints in (4) we know that,

$$\mathbf{s}_{j_i} = \bar{\mathbf{s}}_{l_i},\ \text{for } i = 1,2. \tag{6}$$

On the other hand, for a connected graph there exists a path of nodes $\mathscr{P}$ that connects $l_1, l_2 \in \mathscr{A}$. Moreover, from Def. 1-(b) we know that every sensor $i \in \mathscr{P}$ must have at least two neighbors $l_1', l_2' \in \mathscr{A} \cap \mathscr{P}$, otherwise there would be sensors in $\mathscr{P} \cap \mathscr{A}$ for which there is no sensor in $\mathscr{A}$ at a distance of at most 2 edges from them. We can thus build a path from l_1 to l_2 of the form $l_1 \to i_1 \to l_1' \to i_2 \to l_2' \to \ldots \to l_J' \to i_J \to l_2$, for which $\bar{\mathbf{s}}_{l_1} = \mathbf{s}_{i_1} = \bar{\mathbf{s}}_{l_1'} = \ldots = \bar{\mathbf{s}}_{l_J'} = \mathbf{s}_J = \bar{\mathbf{s}}_{l_2}$. Combining the latter with (6), it follows that $\mathbf{s}_{j_1} = \mathbf{s}_{j_2}$ for arbitrary $j_1, j_2 \in [1,M]$. Thus, any feasible point of (4) is such that $\mathbf{s}_j = \mathbf{s}$, for all $j \in [1,M]$ implying that the arguments of (3) and (4) are equal, which completes the proof. ■

The j-th sensor in (4) is associated with $\mathbf{s}_j \in \mathbb{R}^{p \times 1}$. If l is a bridge sensor, i.e., $l \in \mathscr{A}$, it is also associated with $\bar{\mathbf{s}}_l \in \mathbb{R}^{p \times 1}$. The variables $\{\bar{\mathbf{s}}_l\}_{l \in \mathscr{A}}$ appear only in the constraints of (4) and guarantee that $\mathbf{s}_1 = \ldots = \mathbf{s}_M$ imposing in that way the "consensus" requirement across all the sensors. Different from (3), (4) can be implemented in a distributed fashion as we describe in the next section.

3.1 A coordinate descent algorithm

To solve (4), we will use a coordinate descent algorithm combined with the method of multipliers [1]. Consider the augmented Lagrangian of the optimization problem in (4), that is given by

$$\begin{aligned} \mathscr{L}_a[\mathbf{u}, \bar{\mathbf{s}}, \mathbf{v}] = & \sum_{j=1}^M \|\mathbf{s}_j - M\mathbf{C}_j \mathbf{x}_j\|_2^2 \\ & + \sum_{l \in \mathscr{A}} \sum_{j \in \mathscr{N}_l} (\mathbf{v}_j^l)^T (\mathbf{s}_j - \bar{\mathbf{s}}_l) + \frac{c}{2} \sum_{l \in \mathscr{A}} \sum_{j \in \mathscr{N}_l} \|\mathbf{s}_j - \bar{\mathbf{s}}_l\|_2^2, \end{aligned} \tag{7}$$

where we defined $\mathbf{u} := \{\mathbf{s}_j\}_{j=1}^M$, $\bar{\mathbf{s}} := \{\bar{\mathbf{s}}_l\}_{l \in \mathscr{A}}$ and $\mathbf{v} := \{\mathbf{v}_j^l\}_{j \in [1,M]}^{l \in \mathscr{M}_j}$. The constant $c > 0$ is arbitrary and $\mathbf{v}_j^l$ is the Lagrange multiplier associated with the constraint $\mathbf{s}_j = \bar{\mathbf{s}}_l$, for $j \in [1,M]$ and $l \in \mathscr{M}_j$. The multiplier $\mathbf{v}_j^l$ is located at the j-th sensor.

The process that relies on (7) to yield a distributed implementation is described in the following proposition.

Proposition 2 *Consider the iteration defined for each sensor $j \in [1,M]$ by:*

$$\mathbf{s}_j(k+1) = \frac{1}{2 + c|\mathscr{M}_j|} \left[2M\mathbf{C}_j \mathbf{x}_j - \sum_{l \in \mathscr{M}_j} \mathbf{v}_j^l(k) + c \sum_{l \in \mathscr{M}_j} \bar{\mathbf{s}}_l(k) \right], \tag{8}$$

$$\bar{\mathbf{s}}_l(k+1) = \frac{1}{c|\mathscr{N}_l|} \sum_{j \in \mathscr{N}_l} \mathbf{v}_j^l(k) + \frac{1}{|\mathscr{N}_l|} \sum_{j \in \mathscr{N}_l} \mathbf{s}_j(k+1),\ l \in \mathscr{A}, \tag{9}$$

$$\mathbf{v}_j^l(k+1) = \mathbf{v}_j^l(k) + c[\mathbf{s}_j(k+1) - \bar{\mathbf{s}}_l(k+1)],\ l \in \mathscr{M}_j \tag{10}$$

Algorithm 1 : $\mathbf{C}_j$ known at each sensor

Initialize $\{\mathbf{s}_j(0)\}_{j=1}^M$, $\{\bar{\mathbf{s}}_l(0)\}_{l\in\mathscr{A}}$ and $\{\mathbf{v}_j^l(0)\}_{j=1,\ldots,M}^{l\in\mathscr{M}_j}$ to zero

for $k=0,1,\ldots$ **do**

Bridge sensors $l\in\mathscr{A}$: transmit $\bar{\mathbf{s}}_l(k)$ to its neighbors in $\mathscr{N}_l$

All $j\in[1,M]$: update $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$ by (10)

All $j\in[1,M]$: update $\mathbf{s}_j(k+1)$ using (8).

All $j\in[1,M]$: transmit $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$ to each $l\in\mathscr{M}_j$

Bridge sensors $l\in\mathscr{A}$: compute $\bar{\mathbf{s}}_l(k+1)$ using (9)

end for

where $|\mathscr{M}_j|$ and $|\mathscr{N}_l|$ denote the cardinality of the sets $\mathscr{M}_j$ and $\mathscr{N}_l$ respectively. Then, as $k\to\infty$ the network reaches consensus; i.e.,

$$\lim_{k\to\infty}\mathbf{s}_j(k+1)=\lim_{k\to\infty}\bar{\mathbf{s}}_l(k+1)=\hat{\mathbf{s}},\quad\forall j\in[1,M]. \tag{11}$$

Proof: We wish to show that (8)-(10) generates a succession that converges to the solution of the optimization problem in (4). This follows from using the method of multipliers [1] to minimize the augmented Lagrangian in (7) and update appropriately the corresponding Lagrange multipliers. Let $\mathbf{v}_j^l(k)$ denote the Lagrange multipliers at the k-th iteration. The $(k+1)$-st iteration of the method of multipliers consists of the following two steps:

[S1] Set $\mathbf{v}_j^l=\mathbf{v}_j^l(k)$ and define $\mathbf{v}(k):=\{\mathbf{v}_j^l(k)\}_{j\in[1,M]}^{l\in\mathscr{M}_j}$, to minimize the augmented Lagrangian function in (7) and obtain $\mathbf{s}_j(k+1)$ and $\bar{\mathbf{s}}_l(k+1)$ as:

$$[\mathbf{u}(k+1),\bar{\mathbf{s}}(k+1)]=\arg\min_{\mathbf{u},\bar{\mathbf{s}}}\mathscr{L}_a[\mathbf{u},\bar{\mathbf{s}},\mathbf{v}(k)] \tag{12}$$

with $\bar{\mathbf{s}}(k):=\{\bar{\mathbf{s}}_l(k)\}_{l\in\mathscr{M}_j}$ and $\mathbf{u}(k):=\{\mathbf{s}_j(k)\}_{j\in[1,M]}$

[S2] Update the Lagrange multipliers $\{\mathbf{v}_j^l(k)\}^{l\in\mathscr{M}_j}$ as

$$\mathbf{v}_j^l(k+1)=\mathbf{v}_j^l(k+1)+c[\mathbf{s}_j(k+1)-\bar{\mathbf{s}}_l(k+1)],\ j\in[1,M]. \tag{13}$$

It is known that if $\mathscr{L}_a[\mathbf{u},\bar{\mathbf{s}},\mathbf{v}])$ is the augmented Lagrangian of a convex optimization problem, then [S1]-[S2] converge to the unique global minimum for any constant $c>0$.

Notice though that [S1] requires joint minimization of (7) with respect to $(\mathbf{u},\bar{\mathbf{s}})$ and as such it is not amenable to distributed implementation. A customary turn around is to apply a block coordinate descent method, where we minimize $\mathscr{L}[\mathbf{u},\bar{\mathbf{s}},\mathbf{v}(k)]$ wrt to one variable at a time, effectively replacing [S1] with

[S1-a] For fixed $\mathbf{v}_j^l=\mathbf{v}_j^l(k)$, *and* $\bar{\mathbf{s}}_l=\bar{\mathbf{s}}_l(k)$ minimize (7) wrt $\mathbf{u}$ to obtain

$$\mathbf{u}(k+1)=\arg\min_{\mathbf{u}}\mathscr{L}_a[\mathbf{u},\bar{\mathbf{s}}(k),\mathbf{v}(k)]. \tag{14}$$

Since we have from (7) that the u_j variables are decoupled in $\mathscr{L}_a[\mathbf{u},\bar{\mathbf{s}}(k),\mathbf{v}(k)]$, the optimization in (14) is equivalent to the M separate optimizations

$$\mathbf{s}_j(k+1)=\arg\min_{\mathbf{s}_j}\mathscr{L}_a[\mathbf{u},\bar{\mathbf{s}}(k),\mathbf{v}(k)];\ j\in[1,M]. \tag{15}$$

[S1-b] Setting $\mathbf{s}_j=\mathbf{s}_j(k+1)$ we minimize wrt to $\{\bar{\mathbf{s}}\}$ to obtain

$$\bar{\mathbf{s}}(k+1)=\arg\min_{\bar{\mathbf{s}}}\mathscr{L}_a[\mathbf{u}(k+1),\bar{\mathbf{s}},\mathbf{v}(k)]. \tag{16}$$

As in [S1-a], the $\bar{\mathbf{s}}_l$ variables are decoupled in $\mathscr{L}_a[\mathbf{u}(k+1),\bar{\mathbf{s}},\mathbf{v}(k)]$ and (16) is equivalent to [c.f. (7)]

$$\bar{\mathbf{s}}_l(k+1)=\arg\min_{\bar{\mathbf{s}}_l}\mathscr{L}_a[\mathbf{u}(k+1),\bar{\mathbf{s}},\mathbf{v}(k)];\ l\in\mathscr{A} \tag{17}$$

The algorithm formed by [S1-a], [S1-b], [S2] belongs to the class of the so called alternative multipliers methods which as [S1]-[S2] also converge to the unique global minimum for any constant $c>0$ [1, Chpt. 3]. However, note that (10), (15) and (17) entail local variable updates hinting to the possibility of distributed implementation as we show later; see also Algorithm 1.

To conclude the proof it suffices to show that (8)-(10) are equivalent to (13), (15) and (17). But since the cost functions involved in (15) and (17) are convex, the optimal solution can be obtained by applying the first order optimality conditions

$$\nabla_{\mathbf{s}_j}\mathscr{L}_a[\mathbf{u}(k+1),\bar{\mathbf{s}}(k),\mathbf{v}(k)]=\mathbf{0},\ j\in[1,M] \tag{18}$$

$$\nabla_{\bar{\mathbf{s}}_l}\mathscr{L}_a[\mathbf{u}(k+1),\bar{\mathbf{s}}(k),\mathbf{v}(k)]=\mathbf{0},\ j\in\mathscr{A}. \tag{19}$$

The gradients involved in (18) and (19) can be easily computed after differentiating (7) and are given by:

$$\begin{aligned}\nabla_{\mathbf{s}_j}\mathscr{L}_a&=2\mathbf{s}_j-2M\mathbf{C}_j\mathbf{x}_j+\sum_{l\in\mathscr{M}_j}\mathbf{v}_j^l(k)+c\sum_{l\in\mathscr{M}_j}\left[\mathbf{s}_j-\bar{\mathbf{s}}_l(k)\right],\\ \nabla_{\bar{\mathbf{s}}_l}\mathscr{L}_a&=c\sum_{j\in\mathscr{N}_l}\left[\bar{\mathbf{s}}_l-\mathbf{s}_j(k+1)\right]-\sum_{j\in\mathscr{N}_l}\mathbf{v}_j^l(k).\end{aligned} \tag{20}$$

Substituting (20) into (18) and (19), equations (8) and (9) follow readily; and thus, (10) coincides with (13). ■

The iteration (8)-(10) can be implemented with a distributed algorithm whereby all sensors $j\in[1,M]$ keep track of the local estimate $\mathbf{s}_j(k)$ and the Lagrange multipliers $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$. The sensors that also belong to $\mathscr{A}$ keep track of these variables and the consensus enforcing variables $\bar{\mathbf{s}}_l(k)$. The resulting coordinate descent scheme is summarized as Algorithm 1; see also Fig. 2. At the k-th iteration, sensor j receives the consensus variables $\bar{\mathbf{s}}_l(k)$ from all its bridge neighbors $l\in\mathscr{M}_j$. It uses this information to update its Lagrange multipliers $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$ using (10) and the result to compute $\mathbf{s}_j(k+1)$ through (8). After completing this iteration step, sensor j transmits to each of its bridge neighbors $l\in\mathscr{M}_j$ the vector $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$. The bridge sensor l receives the vectors $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$ from all its neighbors $j\in\mathscr{N}_l$ and proceeds to compute $\bar{\mathbf{s}}_l(k+1)$ using (9). This completes the k-th iteration, and the bridge sensors proceed to transmit $\bar{\mathbf{s}}_l(k+1)$ to all their neighbors $j\in\mathscr{N}_l$ starting the $(k+1)$-st iteration.

During the kth iteration, a sensor $j\in[1,M]$ sends to all its bridge neighbors in $\mathscr{M}_j$ the vectors $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$ $\bar{\mathbf{s}}_l(k)\in\mathbb{R}^{p\times1}$; thus, it transmits $p|\mathscr{M}_j\backslash(j)|$ scalars. Notice, that for a sensor $j\in[1,M]\backslash\mathscr{A}$ it holds that $|\mathscr{M}_j\backslash(j)|=|\mathscr{M}_j|$; but for a sensor $l\in\mathscr{A}$ it holds that $|\mathscr{M}_l\backslash(l)|=|\mathscr{M}_l|-1$, since a bridge sensor l does not have to transmit $\bar{\mathbf{s}}_l(k+1)$ to itself. Based on the previous discussion we can readily infer that each sensor in the WSN has to transmit $p|\mathscr{M}_j|$ scalars per iteration. Thus, the amount of information that each sensor has to communicate per iteration is in the order $O(p)$, which is intuitively reasonable since each sensor wants to compute $\hat{\mathbf{s}}\in\mathbb{R}^{p\times1}$.

4. DISTRIBUTED BLUE

We now consider scenario (s2) in which sensors have available only their corresponding part of the signal model, namely $\mathbf{H}_j$ and $\Sigma_{n_jn_j}$. As in e.g., [7], we assume that the noise is uncorrelated across sensors which means that $\Sigma_{nn}=\mathrm{diag}(\Sigma_{n_1n_1},\ldots,\Sigma_{n_Mn_M})$. The BLUE estimator can then be written as

$$\hat{\mathbf{s}}_{blue}=\left(\sum_{j=1}^M\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{H}_j\right)^{-1}\sum_{j=1}^M\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{x}_j. \tag{21}$$

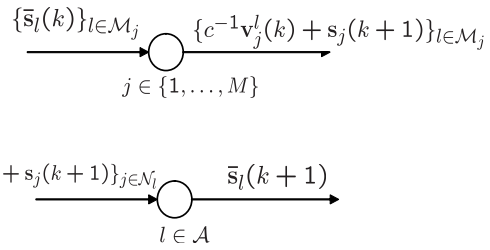

Figure 2: Distributed implementation of Algorithm 1

We assume in (21) that $\mathbf{x}_j$, $\mathbf{H}_j$ and $\Sigma_{n_jn_j}$ are known only to the j-th sensor. In this sense, the estimation algorithm needs not only to disseminate the observations but also the signal model. Even though this problem is seemingly more difficult than the one in Section 3, we will develop a similar algorithm. We start by writing $\hat{\mathbf{s}}_{blue}$ in (21) as the solution of a convex optimization problem in the following proposition[1].

Proposition 3 *The BLUE estimator in* (21) *is given by:*

$$\arg\min_{\mathbf{u}\in\mathbb{R}^{p\times 1}} \sum_{j=1}^{M} \left\|\Sigma_{n_jn_j}^{-1/2}\mathbf{H}_j\mathbf{s}-\Sigma_{n_jn_j}^{-1/2}\mathbf{x}_j\right\|_2^2. \tag{22}$$

Similar to (3), the minimization problem in (22) does not lead to distributed implementation motivating the introduction of the following alternative formulation.

Proposition 4 *If $\mathscr{A}$ is a set of bridge sensors, the minimization problem in (22) is equivalent to*

$$\{\hat{\mathbf{s}}_j\}_{j=1}^M := \arg\min \sum_{j=1}^{M} \left\|\Sigma_{n_jn_j}^{-1/2}\mathbf{H}_j\mathbf{s}_j-\Sigma_{n_jn_j}^{-1/2}\mathbf{x}_j\right\|_2^2$$
$$\text{s.t. } \mathbf{s}_j=\bar{\mathbf{s}}_l,\ l\in\mathscr{A},\ j\in\mathscr{N}_l, \tag{23}$$

in the sense that $\hat{\mathbf{s}}=\hat{\mathbf{s}}_j\ \forall j\in[1,M]$.

The j-th sensor in (23) maintains the local estimate $\mathbf{s}_j$, with the l-th bridge sensors also maintaining $\bar{\mathbf{s}}_l$. The augmented Lagrangian of the optimization problem in (23) can be written as

$$\mathscr{L}_a(\mathbf{u},\bar{\mathbf{s}},\mathbf{v}) = \sum_{j=1}^{M} \left\|\Sigma_{n_jn_j}^{-1/2}\mathbf{H}_j\mathbf{s}_j-\Sigma_{n_jn_j}^{-1/2}\mathbf{x}_j\right\|_2^2 \tag{24}$$
$$+\sum_{l\in\mathscr{A}}\sum_{j\in\mathscr{N}_l}(\mathbf{v}_j^l)^T(\mathbf{s}_j-\bar{\mathbf{s}}_l)+\frac{c}{2}\sum_{l\in\mathscr{A}}\sum_{j\in\mathscr{N}_l}\left\|\mathbf{s}_j-\bar{\mathbf{s}}_l\right\|_2^2.$$

Proceeding as in Section 3.1, we minimize (24) using an alternating multipliers approach to obtain the following proposition.

Proposition 5 *Consider the iteration given by:*

$$\mathbf{s}_j(k+1)=\mathbf{B}_j^{-1}\left(2\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{x}_j-\sum_{l\in\mathscr{M}_j}\mathbf{v}_j^l(k)+c\sum_{l\in\mathscr{M}_j}\bar{\mathbf{s}}_l(k)\right), \tag{25}$$

$$\bar{\mathbf{s}}_l(k+1)=\frac{1}{|\mathscr{N}_l|}\sum_{j\in\mathscr{N}_l}\left[c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)\right],\ l\in\mathscr{A}, \tag{26}$$

$$\mathbf{v}_j^l(k+1)=\mathbf{v}_j^l(k)+c[\mathbf{s}_j(k+1)-\bar{\mathbf{s}}_l(k+1)],\ l\in\mathscr{M}_j, \tag{27}$$

[1]Proofs of claims in this paper can be found in [5].

Algorithm 2 : $\mathbf{H}_j$ and $\Sigma_{n_jn_j}$ known at each sensor

Initialize $\{\mathbf{s}_j(0)\}_{j=1}^M$,$\{\bar{\mathbf{s}}_l(0)\}_{l\in\mathscr{A}}$ and $\{\mathbf{v}_j^l(0)\}_{j=1,\dots,M}^{l\in\mathscr{M}_j}$ to zero.
Perform Cholesky factorization $\{\mathbf{B}_j=\mathbf{L}_j\mathbf{L}_j^T\}_{j=1}^M$.
for $k=0,1,\dots$ **do**
  Bridge sensors $l\in\mathscr{A}$: transmit $\bar{\mathbf{s}}_l(k)$ to its neighbors in $\mathscr{N}_l$;
  All $j\in[1,M]$: update $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$ using (27);
  All $j\in[1,M]$: update $\mathbf{s}_j(k+1)$ using (25);
  All $j\in[1,M]$: transmit $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$ to each $l\in\mathscr{M}_j$;
  Bridge sensors $l\in\mathscr{A}$: compute $\bar{\mathbf{s}}_l(k+1)$ by solving lower and upper triangular systems in (29) and (30).
end for

with $\mathbf{B}_j=2\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{H}_j+c|\mathscr{M}_j|\mathbf{I}_p$. *Then, as $k\to\infty$ the network reaches consensus in the sense that*

$$\lim_{k\to\infty}\mathbf{s}_j(k+1)=\lim_{k\to\infty}\bar{\mathbf{s}}_l(k+1)=\hat{\mathbf{s}}_{blue},\ \forall j\in[1,M]. \tag{28}$$

From (25)-(27) we obtain the distributed Algorithm 2. Interestingly, the matrix inversion in (25) can be avoided. Indeed, the matrix $\mathbf{B}_j$ is time invariant, symmetric and positive definite because $|\mathscr{M}_j|>0$ for $j\in[1,M]$. Thus, the jth sensor can perform, during the start-up period of the WSN, Cholesky factorization of $\mathbf{B}_j$, to find a lower triangular matrix $\mathbf{L}_j$ such that $\mathbf{B}_j=\mathbf{L}_j\mathbf{L}_j^T$. Then during iteration k of the distributed algorithm, $\mathbf{u}_j(k+1)$ can be computed by solving the lower triangular system

$$\mathbf{L}_j\mathbf{z}_j(k+1)=2\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{x}_j-\sum_{l\in\mathscr{M}_j}\mathbf{v}_j^l(k)+c\sum_{l\in\mathscr{M}_j}\bar{\mathbf{s}}_l(k), \tag{29}$$

and then obtaining $\mathbf{s}_j(k+1)$ with backward substitution in the upper triangular system

$$\mathbf{L}_j^T\mathbf{s}_j(k+1)=\mathbf{z}_j(k+1). \tag{30}$$

The computational cost for obtaining $\mathbf{s}_j(k+1)$ is thus $O(p^2)$.

Since each sensor $j\in[1,M]$ has available the vector $\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{x}_j$ and the Lagrange multipliers $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$, and it receives the consensus variables $\bar{\mathbf{s}}_l(k)$ from all its bridge sensor neighbors $l\in\mathscr{M}_j$ it is able to update the Lagrange multipliers $\{\mathbf{v}_j^l(k)\}_{l\in\mathscr{M}_j}$ through (27) and compute $\mathbf{s}_j(k+1)$ using (25). Next, sensor j transmits to all its bridge neighbors the vectors $c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)$ for $l\in\mathscr{M}_j$. Then, every sensor $l\in\mathscr{A}$ receives the vectors $\{c^{-1}\mathbf{v}_j^l(k)+\mathbf{s}_j(k+1)\}_{l\in\mathscr{N}_l}$ and forms $\bar{\mathbf{s}}_l(k+1)$ through (26). The communication cost for each sensor is as in Section 3.1, i.e, $p|\mathscr{M}_j|$ scalars per sensor.

Remark 1 Another scheme for computing the BLUE estimator in a distributed fashion was developed in [7] where separate consensus algorithms are run to determine the matrix $\mathbf{F}_{blue}=\sum_{j=1}^M\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{H}_j$ and vector $\mathbf{f}_{blue}=\sum_{j=1}^M\mathbf{H}_j^T\Sigma_{n_jn_j}^{-1}\mathbf{x}_j$. With respect to [7], our method has the same computational complexity $O(p^2)$ while the communication cost is reduced from $O(p^2)$ for the method in [7] to $O(p)$ for Algorithm 2. Also, as we verify in Section 5 our can approach exhibit a considerably faster convergence ratio.

5. NUMERICAL RESULTS

In this section, we provide numerical results comparing the performance of Algorithm 2 in Section 4 against the consensus averaging

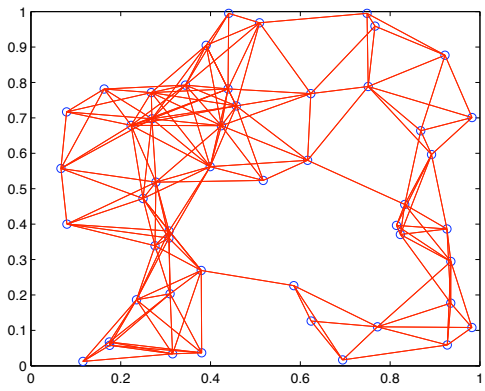

Figure 3: Randomly generated 50-sensor WSN.

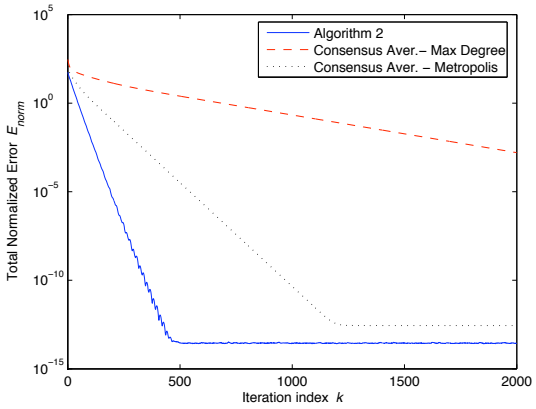

Figure 4: Normalized total error vs iteration index k for Algorithm 2 and the consensus averaging scheme in [7].

scheme in [7]. The metric used for comparison is the Euclidean norm of the error between the local estimates and the BLUE estimate. The total normalized error is thus given by

$$E_{norm}(k) = \sum_{j=1}^{M} \frac{\|\mathbf{s}_j(k) - \hat{\mathbf{s}}_{blue}\|^2}{\|\hat{\mathbf{s}}_{blue}\|^2}, \qquad (31)$$

where $\mathbf{s}_j(k)$ is the local estimate at the jth sensor for the k-th iteration.

We consider a sensor network consisting of 50 sensors; see Fig. 3. The WSN is generated by randomly placing nodes according to a uniform distribution in the unit square $[0,1] \times [0,1]$. We assume that two sensors are able to communicate – and are thus connected with an edge in Fig. 3 – if their Euclidean distance is less than $1/4$. With respect to the signal model we assume that each sensor has 10 observations; i.e., $L_1 = \ldots = L_{50} = 10$, and that $\mathbf{s}$ incorporates $p = 5$ parameters. The entries of the observation matrices $\{\mathbf{H}_j\}_{j=1}^{50}$ contain independent random variables uniformly distributed in the set $[-0.5, 0.5]$. The noise is unit power white Gaussian implying that $\mathbf{\Sigma}_{n_j n_j} = \mathbf{I}_{10\times 10}$, for $j = 1, \ldots, 50$. The bridge sensor set $\mathscr{A}$ contains sensors selected such that $|\mathscr{M}_j| \leq 5 = p$ for $j = 1, \ldots, 50$. The parameter c in Algorithm 2 is set equal to 6. For the algorithm in [7], we test both the max-degree and the Metropolis weights.

In Fig. 4 we plot the normalized total error [c.f. (31)] versus iteration index. We compare the convergence rate of Algorithm 2 with the rate exhibited by the consensus averaging approach in [7] using the max-degree and Metropolis weights. Clearly, Algorithm 2 outperforms the consensus averaging approach. Also, recall that the communication costs per iteration in our approach have been reduced, since each sensor has to transmit $|\mathscr{M}_j|p$ scalars with $|\mathscr{M}_j| \leq p$ compared to the consensus averaging approach. Intuitively, presence of the consensus constraints in (23) and the utilization of the "consensus" variables $\bar{\mathbf{s}}_l$ for $l \in \mathscr{A}$ help the sensors to reach agreement faster. The choice of the constant c is important for the speed of convergence of the algorithm and future work will consider ways of determining c based solely on local information.

6. CONCLUDING REMARKS

We developed distributed algorithms for estimation using ad hoc WSNs based on successive refinement of local estimates. Per transmission cycle, information is communicated to one-hop neighbors only; the information received from these neighbors is then used to improve the local estimate. The approach follows after expressing the best linear unbiased estimator (BLUE) as the solution of judiciously designed convex optimization problems, and relies on the method of multipliers, coordinate descent and other optimization techniques to enable parallel/distributed implementation. The resultant algorithms are guaranteed to converge to the (optimal) BLUE estimate. Numerical results corroborated the asserted convergence claims and indicated that our algorithms attain faster convergence rates than existing alternatives.

Future research topics include generalizing our approach to estimation of random parameters as well as a better understanding of the convergence rate.[2]

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Second Edition, 1999

[2] V. Delouille, R. Neelamani and R. Baraniuk, "Robust Distributed Estimation in Sensor Networks using the Embedded Polygons Algorithm," *Proc. of the 3rd Intl. Symp. on Info. Proc. in Sensor Networks*, Berkeley, CA, April 2004.

[3] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[4] M. G. Rabbat, R. D. Nowak and J. A. Bucklew, "Generalized Consensus Computation in Networked Systems With Erasure Links," *Proc. of the 6th Workshop on Sig. Processing Advances in Wireless Communications*, pp. 1088–1092, New York, NY, June 5-8 2005.

[5] I. D. Schizas, A. Ribeiro, G. B. Giannakis, "Distributed Estimation in Ad-Hoc Wireless Sensor Networks", in preparation.

[6] D. P. Spanos, R. O. Saber and R. M. Murray, "Distributed Sensor Fusion Using Dynamic Consensus", *Proc. of the 16th IFAC World Congress*, Prague, July 2005.

[7] L. Xiao, S. Boyd and S. Lall, "A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus," *Proc. of the 4th Intl. Symp. on Info. Processing in Sensor Networks*, pp. 63–70, Berkeley, CA, April 2005.

[8] L. Xiao, S. Boyd, "Fast Linear Iterations for Distributed Averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[2] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the Army Research Laboratory or the U. S. Government.