A Decentralized Framework for Linear Coherent Estimation with Spatial Collaboration

Swarnendu Kar
New Devices Group, Intel Corporation,
Hillsboro, OR, 97124, USA
Email: swkar@syr.edu

Pramod K. Varshney
Dept. of EECS, Syracuse University,
Syracuse, NY, 13244, USA
Email: varshney@syr.edu

***Abstract*—We study an estimation problem where a fusion center estimates a random parameter by using a partially connected network of sensor nodes. The process involves two stages. In the collaboration stage, the sensor nodes share their observations with their neighbors. In the estimation stage, all the sensor nodes form a coherent beam to the fusion center using the analog amplify-and-forward procedure. In our previous work on this topic [1], a control center determines the optimum collaboration strategy that is sent to the sensor nodes prior to starting the two-stage procedure. In this paper, we develop a new framework where the collaboration strategies are computed in a decentralized manner using minimal communication with the control center. This makes the sensor network more energy efficient and reduces control channel communication requirements.**

I. Introduction

For estimation applications in a wireless sensor network, multiple sensor nodes send their observations in a coordinated manner to a fusion center (FC) – so as to conserve energy resources while achieving the desired accuracy of estimation. Several frameworks (analog transmissions [2] vs quantization based [3]) and protocols (distributed [4] vs collaborative [5]) have been considered by the researchers so far. In this paper, we consider the analog amplify-and-forward framework for transmission, which is widely used in the literature [2],[4] due to its simplicity in implementation and provably optimal information theoretic properties for simple networks [6].

Recent studies have indicated that sharing observations among neighboring nodes prior to communication with FC, a protocol termed as collaborative estimation [5],[1], can save significant energy resources compared to distributed estimation, where no in-network communication is permitted [3],[2]. In [1], we derived the optimum energy-constrained collaboration strategy – which provides the precise weights that each node must use to combine and amplify their neighbors' observations, prior to transmission to the FC. This paper is a continuation of [1] where we switch our focus from optimality of the strategy to the other important aspect of computability.

While the optimal collaboration strategy guarantees energy efficiency, it also poses a coordination challenge. Since the collaboration weights are obtained by solving an optimization problem based on a specific accuracy constraint (or equivalently, transmission energy constraint, these two quantities represent a tradeoff), it turns out that higher accuracy cannot be achieved by simply scaling the weights of a previous strategy that was optimized for a lower-accuracy requirement. Consequently, each time the accuracy requirement of the application changes, the corresponding optimal strategies must be centrally computed at some control center (CC) (possibly the FC itself) and relayed back to the constituent nodes through a reliable and separate control channel. In this paper, we seek to alleviate this concern by exploring decentralized approaches for computing the collaboration strategy.

Under our proposed methodology, the CC broadcasts only one quantity (the cumulative energy to be used at a given time) to all the sensor nodes, based on the accuracy requirements of the particular estimation session. Based on the cumulative energy, the sensors communicate among themselves using only local interactions in order to compute a *reasonably efficient* collaboration strategy. This strategy is subsequently used by the nodes to combine/amplify their neighbors' observations during transmission to the FC. Since the optimal collaboration strategy in [1] is not amenable for decentralized computation, we explore three different strategies that can be computed in the network – the first two strategies are efficient in the low and high-energy regimes respectively and the third strategy, which interpolates between the first two strategies, is reasonably efficient across the entire energy regime. We compare the efficiency of the aforementioned strategies vis-a-vis the optimal strategy. Due to the decentralized computation of collaborative strategies, the network requires less coordination with the CC and adapts faster to changes in application requirements.

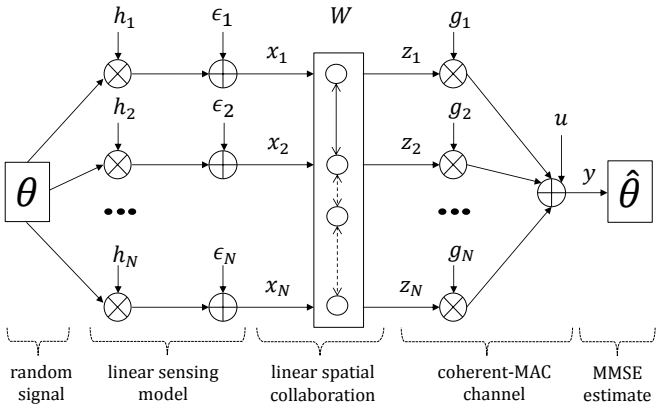

Fig. 1. Wireless sensor network performing collaborative estimation.

II. Problem Formulation

The estimation problem is depicted in Figure 1. The random parameter to be estimated, θ, is assumed to be a zero-mean Gaussian with variance η^2. Different noisy versions of θ are observed by N sensors. The observation vector is $\boldsymbol{x} = [x_1, \ldots, x_N]$ where $x_n = h_n\theta + \epsilon_n$, with h_n and ϵ_n denoting the observation gain and measurement noise respectively. The measurement noise variables $\{\epsilon_n\}_{n=1}^N$ are assumed to be independent and identically distributed (iid) Gaussian random variables with zero mean and variance σ^2.

The availability of collaboration links is represented by the symmetric adjacency matrix $\boldsymbol{A}$, where $A_{nm} = 1$ (or $A_{nm} = 0$) implies that node n has (or does not have) access to the observation of node m. Corresponding to an adjacency matrix $\boldsymbol{A}$ and an $\boldsymbol{A}$-sparse[1] matrix $\boldsymbol{W}$, *collaboration* is defined as individual nodes being able to

This research was supported by the National Science Foundation under Grant No. 0925854.

[1]Define an $\boldsymbol{A}$-sparse matrix as one for which non-zero elements may appear only at locations (n, m) for which $A_{nm} = 1$. The set of all $\boldsymbol{A}$-sparse matrices is denoted by $\mathcal{S}_A$.

linearly combine local observations from other collaborating nodes $z_n = \sum_{m\in\mathcal{B}_n} \boldsymbol{W}_{nm} x_m$, where $\mathcal{B}_n \triangleq \{m:\ \boldsymbol{A}_{nm} = 1\}$, without any further loss of information. In effect, the network is able to compute a one-shot spatial transformation of the form $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}$. In practice, this transformation is realizable when any two neighboring sensors are close enough to ensure reliable information exchange.

The transformed observations $\{z_n\}_{n=1}^N$ are transmitted to the FC through a coherent MAC channel, so that the received signal is $y = \boldsymbol{g}^T\boldsymbol{z} + u$, where $\boldsymbol{g}$ and u describe the channel gains and the channel noise respectively. The channel noise u is assumed to be Gaussian distributed with zero mean and variance ξ^2. The FC receives the noise-corrupted signal y and computes an estimate of θ. Since y is a linear Gaussian random variable conditioned on θ,

$$\begin{aligned} &\theta \sim \mathcal{N}(0, \eta^2), \text{ and} \\ &y|\theta \sim \mathcal{N}\left(\underbrace{\boldsymbol{g}^T\boldsymbol{W}\boldsymbol{h}}_{\triangleq\mu \text{ (net gain)}}\ \theta,\ \underbrace{\boldsymbol{g}^T\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^T\boldsymbol{g} + \xi^2}_{\triangleq\zeta^2 \text{ (net noise variance)}} \right), \end{aligned} \tag{1}$$

the minimum-mean-square-error (MMSE) estimator $\widehat{\theta} = \mathbb{E}[\theta|y]$ is the optimal fusion rule. From estimation theory (for details the reader is referred to [7]), the MMSE estimator and resulting distortion $D_{\boldsymbol{W}}$ are given by

$$\widehat{\theta} = \frac{1}{1 + \frac{\zeta^2}{\eta^2\mu^2}} \frac{y}{\mu},\ \text{and}\ \frac{1}{D_{\boldsymbol{W}}} = \frac{1}{\eta^2} + J_{\boldsymbol{W}},\ J_{\boldsymbol{W}} = \frac{\mu^2}{\zeta^2}, \tag{2}$$

where the quantity $J_{\boldsymbol{W}}$ is the Fisher Information and μ and ζ^2 are the net gain and net noise variance as defined in Equation (1). The cumulative transmission energy required to transmit the transformed observations $\boldsymbol{z}$ is

$$\begin{aligned} &\mathcal{E}_{\boldsymbol{W}} = \mathbb{E}[\boldsymbol{z}^T\boldsymbol{z}] = \text{Tr}\ \left[\boldsymbol{W}\boldsymbol{E}_{\mathsf{x}}\boldsymbol{W}^T\right],\ \text{where} \\ &\boldsymbol{E}_{\mathsf{x}} \triangleq \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \eta^2\boldsymbol{h}\boldsymbol{h}^T + \boldsymbol{\Sigma}. \end{aligned} \tag{3}$$

Note that the quantities μ, ζ^2 and, therefore, the distortion D (equivalently J) and also the required energy $\mathcal{E}$ depend on the choice of the collaboration matrix $\boldsymbol{W}$. To enable efficient resource allocation, the CC can compute the optimal collaboration matrix subject to a cumulative transmission energy constraint

$$\boldsymbol{W}_{\text{opt}} = \arg\min_{\boldsymbol{W}\in\mathcal{S}_A} D_{\boldsymbol{W}},\ \text{s.t.}\ \mathcal{E}_{\boldsymbol{W}} \le \mathcal{E}, \tag{4}$$

and communicate the corresponding weights $\boldsymbol{W}_{\text{opt}}$ to the sensor nodes via a separate and reliable control channel. The exact form of $\boldsymbol{W}_{\text{opt}}$ and corresponding J_{opt} were derived in [1] and are summarized below. Here, L is the cardinality of $\boldsymbol{A}$, which is also the number of non-zero collaboration weights. In an equivalent representation, we construct $\boldsymbol{w} \in \mathbb{R}^L$ by concatenating those elements of $\boldsymbol{W}$ that are allowed to be non-zero. Accordingly, we define the $L \times L$ matrix $\boldsymbol{\Omega}$ and $L \times N$ matrix $\boldsymbol{G}$ such that the identities

$$\text{Tr}\ \left[\boldsymbol{W}\boldsymbol{E}_{\mathsf{x}}\boldsymbol{W}^T\right] = \boldsymbol{w}^T\boldsymbol{\Omega}\boldsymbol{w},\ \text{and}\ \boldsymbol{g}^T\boldsymbol{W} = \boldsymbol{w}^T\boldsymbol{G}, \tag{5}$$

are satisfied. The following result holds.

Theorem 1 (Optimal single-snapshot estimation, [1]): The optimal Fisher Information is,

$$\begin{aligned} &J_{\text{opt}} = \boldsymbol{h}^T\left(\boldsymbol{\Sigma} + \boldsymbol{\Gamma}/\mathcal{E}_\xi\right)^{-1}\boldsymbol{h},\ \text{where} \\ &\mathcal{E}_\xi \triangleq \mathcal{E}/\xi^2,\ \text{and}\ \boldsymbol{\Gamma} \triangleq \left(\boldsymbol{G}^T\boldsymbol{\Omega}^{-1}\boldsymbol{G}\right)^{-1}, \end{aligned} \tag{6}$$

which is achieved when the collaboration weights are

$$\boldsymbol{w} = \kappa\boldsymbol{w}_{\text{opt}},\ \boldsymbol{w}_{\text{opt}} = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\left(\boldsymbol{\Sigma} + \boldsymbol{\Gamma}/\mathcal{E}_\xi\right)^{-1}\boldsymbol{h}, \tag{7}$$

with the scalar κ chosen to satisfy $\boldsymbol{w}^T\boldsymbol{\Omega}\boldsymbol{w} = \mathcal{E}$.

Though (7) gives the optimal collaboration strategy, it cannot be computed in the network in a decentralized manner. In this paper, we look for strategies that can be computed by the sensor nodes with the help of local interactions alone. Towards that end and inspired by the optimal strategy (7), we evaluate two simpler strategies that are optimal in specific energy regimes. In the low energy regime ($\mathcal{E} \to 0$), we have $\boldsymbol{w}_{\text{opt}} \propto \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{h} =: \boldsymbol{w}_L$, while in the high energy regime ($\mathcal{E} \to \infty$), we have $\boldsymbol{w}_{\text{opt}} \propto \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}\boldsymbol{h} =: w_H$. We next ask the question – are strategies $\boldsymbol{w}_L$ and $\boldsymbol{w}_H$ amenable for decentralized computation? This paper answers this question in the affirmative.

While the strategies $\boldsymbol{w}_L$ and $\boldsymbol{w}_H$ are optimal in the low-energy and high-energy regimes, it turns out that they are also highly sub-optimal beyond those regimes. This implies that neither $\boldsymbol{w}_L$ nor $\boldsymbol{w}_H$ is effective when the energy regime is subject to change in a particular deployment. This leads us to the search for *universal* strategies that are effective across the *all* energy regimes. As a potential candidate, we evaluate a strategy that linearly interpolates between $\boldsymbol{w}_L$ and $\boldsymbol{w}_H$, in particular, $\boldsymbol{w}_U = \alpha\boldsymbol{w}_L + (1-\alpha)\boldsymbol{w}_H$, where α is chosen based on the particular energy regime. Intuitively, α should decrease from 1 to 0 as the operational regime changes from low-energy to high-energy. Also, once α is provided, we can clearly compute $\boldsymbol{w}_U$ in a decentralized manner. So the next questions is – what is an effective way to select α such that the value of α itself can be computed in a decentralized manner? Specifically, can the optimal value

$$\arg\min_{\alpha} D_{\boldsymbol{W}},\ \ \text{s.t.}\ \boldsymbol{w} \propto \alpha\boldsymbol{w}_L + (1-\alpha)\boldsymbol{w}_H,\ \mathcal{E}_{\boldsymbol{W}} \le \mathcal{E}, \tag{8}$$

be computed in a decentralized manner? This paper answers this question in the affirmative as well.

III. Main Results

A. Optimal interpolation for universal strategy

Before we focus on the decentralized computational aspects, we summarize all the strategies discussed so far,

$$\begin{aligned} &\text{(Optimal)} && \boldsymbol{w}_{\text{opt}} = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\left(\boldsymbol{\Sigma} + \boldsymbol{\Gamma}/\mathcal{E}_\xi\right)^{-1}\boldsymbol{h} \\ &\text{(Low-energy)} && \boldsymbol{w}_L = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{h} \\ &\text{(High-energy)} && \boldsymbol{w}_H = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}\boldsymbol{h} \\ &\text{(Universal)} && \boldsymbol{w}_U = \alpha_U\boldsymbol{w}_L + (1-\alpha_U)\,\boldsymbol{w}_H. \end{aligned} \tag{9}$$

The first three strategies are well defined. The quantity α_U in the universal strategy results from solving the optimal interpolation problem posed in (8).

Proposition 2 (Universal strategy): The solution to problem (8) is given by

$$\begin{aligned} &\alpha_U = \frac{1}{1 + \mathcal{E}_\xi c_4},\ \text{where}\ \mathcal{E}_\xi \triangleq \frac{\mathcal{E}}{\xi^2},\ c_4 \triangleq \frac{c_1^2 - c_2 J_0}{J_0^2 - c_1 c_3}, \\ &J_0 \triangleq \boldsymbol{h}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{h},\ c_1 \triangleq \boldsymbol{h}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{h}, \\ &c_2 \triangleq \boldsymbol{h}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Gamma}^{-1}\boldsymbol{h}\ \text{and}\ c_3 = \boldsymbol{h}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}\boldsymbol{h}. \end{aligned} \tag{10}$$

The proof of Proposition 2 is relegated to a future extended version of this paper. It is worth reminding that implementation of all the strategies in (9) involves appropriate scaling factors κ so as to satisfy $\boldsymbol{w}^T\boldsymbol{\Omega}\boldsymbol{w} = \mathcal{E}$. So clearly, κ depends on the operational regime $\mathcal{E}$. Since $\boldsymbol{w}_L$ and $\boldsymbol{w}_H$ are independent of $\mathcal{E}$, it is possible to switch operational regimes for the low-energy and high-energy strategies just by recomputing the scaling factor κ. However, this is not the case with the optimal and universal strategies – for $\boldsymbol{w}_{\text{opt}}$, the entire vector needs to be recomputed when $\mathcal{E}$ needs changing (hence the difficulty in decentralized implementation) while for $\boldsymbol{w}_U$, only one other additional scalar term α_U needs to be recomputed.

B. Performance analysis through numerical simulations

Next, we perform a numerical experiment to compare the performance of various strategies in (9) with that of the optimal strategy. We simulate a random geometric graph of $N = 20$ nodes, with nodes placed randomly in a unit square and collaborative links available for node pairs within distance $r = 0.3$. The observation and channel gains are random numbers chosen uniformly in the range $h_n \sim \mathcal{U}(0,1), g_n \sim \mathcal{U}(0,1)$. The observation noise variance was chosen in the range $\sigma_n^2 \sim \mathcal{U}(0.5, 1.5)$ and channel noise variance set $\xi^2 = 1$. The variance of θ is chosen $\eta^2 = 0.1$. With these parameters, the minimum achievable distortion is $D_0 = 1/(1/\eta^2 + J_0) \approx 0.062$. We simulated the entire operational region by selecting 1000 uniformly spaced values for $D \in (D_0, \eta^2)$ – which was wide enough to cover both low-energy (high-distortion, $D \to \eta^2$) and high-energy (low-distortion, $D \to D_0$) regimes, as depicted in Figure 2. For each distortion value, we calculate the corresponding transmission energy required under all four strategies. Since the optimal strategy yields the lowest transmission energy by definition, we normalize the other three energy values and plot the relative inefficiency value $(\mathcal{E}(D)/\mathcal{E}_{\mathsf{opt}}(D) - 1)$ in terms of percentage. We observe from Figure 2 that the low and high-energy strategies, though optimal in their respective regimes can become highly suboptimal at the opposite ends of their intended operating region – confirming the notion that these strategies are not universal. In particular, the low-energy strategy performs quite poorly in the vicinity of a distortion threshold (marked D_{low}) – below which it ceases to function at all, no matter how much energy is used for transmission. This threshold is actually $\lim_{\kappa\to\infty} D_{\boldsymbol{W}}$ with $\boldsymbol{w} = \kappa \boldsymbol{w}_L$, which is equal to $D_{\mathsf{low}} = 1/(1/\eta^2 + c_1^2/c_2) \approx 0.067$. As expected, the universal strategy performs consistently better than both the low and high-energy strategies in the entire operational region. For this particular example, the universal strategy seems to perform as well as the optimal strategy for all distortion requirements. The inefficiency plot is seen to peak near 1% before dropping off further towards the edges.

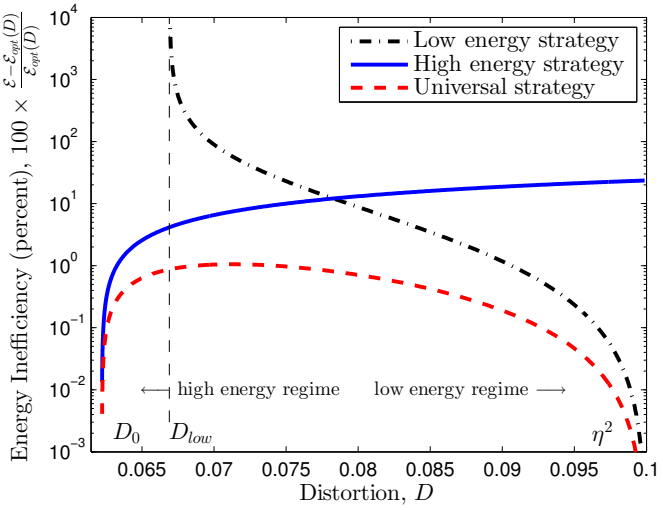

Fig. 2. Efficiency of low-energy, high-energy and universal strategies

C. Decentralized computation

Decentralized message sharing protocols enable the computation of different kinds of functions in a network. In this paper, we make use of two such procedures, namely 1) distributed average consensus (AC) and 2) optimization using alternating direction method of multipliers (ADMM). While we refer the reader to [8] and [9]-(Chapter 3) respectively for detailed discussions, we summarize the key results below concerning AC and ADMM. The variable t represents successive iterations.

Proposition 3 (Average consensus, [8]): Consider the problem to compute the average $p_{\mathsf{avg}} = \frac{1}{N}\mathbf{1}^T\boldsymbol{p}(0)$ at all nodes, where $p_n(0)$ denotes a value available only to the nth node at the beginning of iterations. Consider a matrix $\boldsymbol{Q}$ with elements Q_{nm} such that $\boldsymbol{p}(t+1) = \boldsymbol{Q}\boldsymbol{p}(t)$ represent distributed linear iterations of the form

$$p_n(t+1) = \sum_{m\in\mathcal{B}_n} Q_{nm} p_m(t),\ n = 1, \cdots, N, \tag{11}$$

where $\mathcal{B}_n$ denotes the neighborhood set of node n (including itself). Choose an appropriately sparse $\boldsymbol{Q}$ such that $\lim_{t\to\infty}\boldsymbol{Q}^t = \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Then $\lim_{t\to\infty}\boldsymbol{p}(t) = p_{\mathsf{avg}}\mathbf{1}$, i.e., $\lim_{t\to\infty} p_n(t) = p_{\mathsf{avg}}$ for all nodes.

When pre-scaled by N, AC can be used to compute the sum of the initial values rather than the mean. We will use AC to compute several sums in this paper.

Proposition 4 (ADMM, [9]-(Chapter 3)): Consider the following optimization problem with separable convex objectives functions $F_n : \mathcal{R}^{d_n} \to \mathcal{R}$

$$\min \sum_{n=1}^{N} F_n(\boldsymbol{p}_n),\ \text{ s.t. } \boldsymbol{e}_j^T\boldsymbol{p} = s_j,\ j = 1, \cdots, J, \tag{12}$$

where $\boldsymbol{p} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_N)$ and $\boldsymbol{p}_n$ denotes a subvector of dimension d_n. Let vectors $\boldsymbol{e}_{jn}$ denote the subvector of $\boldsymbol{e}_j$ that corresponds to $\boldsymbol{p}_n$, let $\mathcal{I}(j)$ be the set of indices n of subvectors $\boldsymbol{p}_n$ that appear in the jth constraint $\boldsymbol{e}_j^T\boldsymbol{p} = s_j$, i.e. $\mathcal{I}(j) = \{n | \boldsymbol{e}_{jn} \neq 0\}$. Then problem (12) can be solved exactly in a distributed manner through the following iterations

$$\boldsymbol{p}_n(t+1) = \arg\min_{\boldsymbol{p}_n} \Bigg\{ F_n(\boldsymbol{p}_n) + \sum_{j|n\in\mathcal{I}(j)} \Big\{ \lambda_j(t)\boldsymbol{e}_{jn}^T\boldsymbol{p}_n + \frac{c}{2}\Big(\boldsymbol{e}_{jn}^T(\boldsymbol{p}_n - \boldsymbol{p}_n(t)) + \boldsymbol{r}_j(t)\Big)^2 \Big\} \Bigg\}, \tag{13a}$$

$$\lambda_j(t+1) = \lambda_j(t) + c\boldsymbol{r}_j(t+1),\ j = 1, \cdots, J, \tag{13b}$$

$$\boldsymbol{r}_j(t) \triangleq \frac{1}{|\mathcal{I}(j)|}\Big(\boldsymbol{e}_j^T\boldsymbol{p}(t) - s_j\Big). \tag{13c}$$

The initial vectors $\boldsymbol{p}(0)$ and $\lambda(0)$ can be arbitrary. The constant c can be chosen via experiments to result in a desired convergence speed.

We now describe how to use the consensus and ADMM algorithms to compute the three – low-energy $\boldsymbol{w}_L$, high-energy $\boldsymbol{w}_H$ and universal $\boldsymbol{w}_U$ strategies, along with the appropriate scaling factors κ, in a decentralized manner. It is assumed that, based on the quality requirements, the cumulative energy $\mathcal{E}$ is known (broadcast by CC) to all the individual nodes. In this paper, we ignore the cost of local communications related to consensus/ADMM procedures and assume that the strategies are computed without any errors. For a detailed exposition on potentially lossy local communications in consensus algorithms, the reader is referred to [10] and references therein.

Notations: In order to define $\boldsymbol{\Omega}$ and $\boldsymbol{G}$ in Proposition 1, we concatenate the elements of $\boldsymbol{W}$ column-wise (only those that are allowed to be non-zero), and construct[2] $\boldsymbol{w} = (\boldsymbol{w}_{\mathcal{T}_1}, \cdots, \boldsymbol{w}_{\mathcal{T}_N}) = [w_1, w_2, \ldots, w_L]^T$. The matrix $\boldsymbol{\Omega}$ is formed from $\boldsymbol{E}_{\mathsf{x}}$ by assigning the submatrices $\boldsymbol{E}_{\mathsf{x},\mathcal{B}_n}$ to $\boldsymbol{\Omega}_{\mathcal{F}_n}$ for all n. The matrix $\boldsymbol{G}$ is formed from vector $\boldsymbol{g}$ by assigning the subvector $\boldsymbol{g}_{\mathcal{B}_n}$ to $\boldsymbol{G}_{\mathcal{T}_n,n}$ for all n. We illustrate these notations through an example, in Figure 3, with $N = 3$ nodes and 2 collaborating links, i.e., total $L = 7$ non-zero coefficients.

[2]Subvector $\mathcal{B}_n$ denotes the neighbors of node n, ordered, including itself. Subvectors $\mathcal{T}_n$ and $\mathcal{F}_n$ denote the $\boldsymbol{w}$-indices such that $\mathcal{T}_n$ forms the nth column of $\boldsymbol{W}$ and $\mathcal{F}_n$ forms the nth row of $\boldsymbol{W}$. See Figure 3 for example.

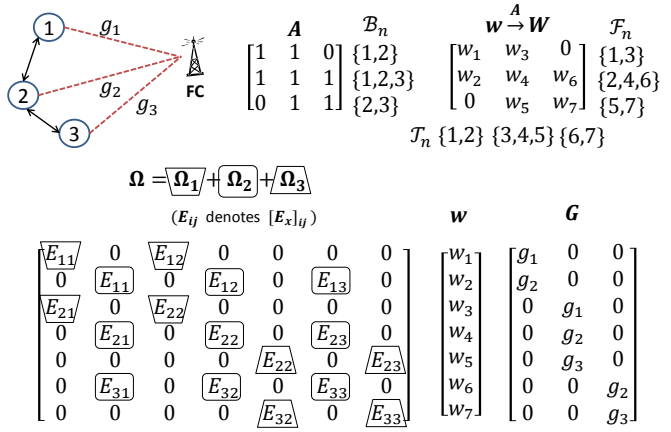

Fig. 3. Transformation from $\boldsymbol{W}$ to $\boldsymbol{w}$ and associated matrices $\boldsymbol{\Omega}$ and $\boldsymbol{G}$.

D. Low-energy strategy

Distributed computation of $\kappa_L \boldsymbol{w}_L$, i.e., the vector aligned to $\boldsymbol{w}_L = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{h}$ and scaled such that the cumulative energy is $\mathcal{E}$, is particularly simple. It turns out (refer to Figure 3 for an illustration of notations) that the subvector $\boldsymbol{w}_{L,\mathcal{F}_n}$ reduces to $g_n \boldsymbol{\Omega}_{\mathcal{F}_n}^{-1} \boldsymbol{h}_{\mathcal{B}_n}$, which can be computed independently by node n. The scaling factor κ_L is determined by computing $\boldsymbol{w}_L^T \boldsymbol{\Omega} \boldsymbol{w}_L$ through consensus with a starting value of $\boldsymbol{w}_{L,\mathcal{F}_n}^T \boldsymbol{\Omega}_{\mathcal{F}_n} \boldsymbol{w}_{L,\mathcal{F}_n}$ at each node. Figure 4 illustrates the aforementioned steps.

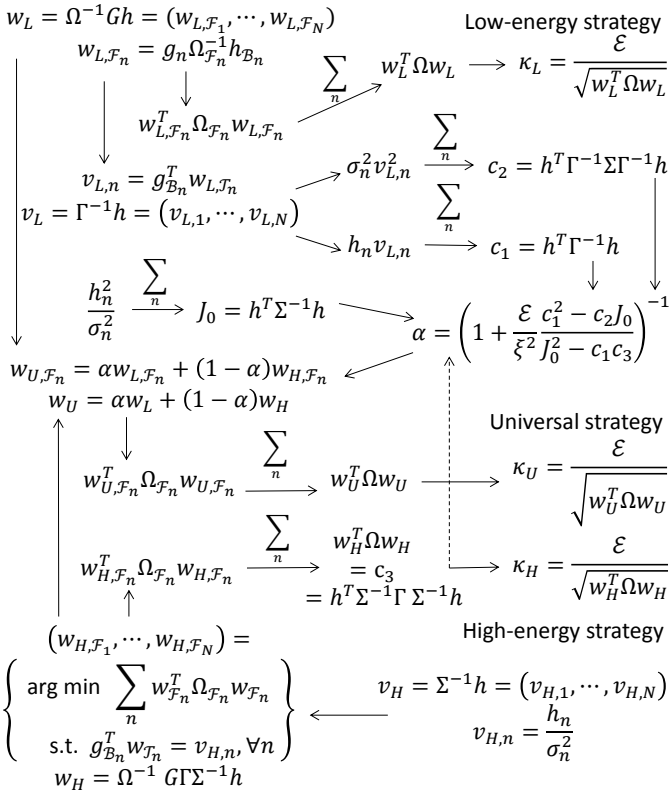

Fig. 4. Decentralized computation of low-energy, high-energy and universal strategies

E. High-energy strategy

In order to compute $\boldsymbol{w}_H = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}\boldsymbol{h}$, we use the following well-known result concerning the minimization of a convex quadratic objective function subject to linear constraints,

$$\left\{ \arg\max_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{\Omega} \boldsymbol{w}, \text{ s.t. } \boldsymbol{G}^T \boldsymbol{w} = \boldsymbol{v} \right\} = \boldsymbol{\Omega}^{-1}\boldsymbol{G}\boldsymbol{\Gamma}\boldsymbol{v}, \qquad (14)$$

where $\boldsymbol{\Omega}$ is positive definite and $\boldsymbol{\Gamma} = \left(\boldsymbol{G}^T \boldsymbol{\Omega}^{-1} \boldsymbol{G}\right)^{-1}$ is assumed to exist. In [11], an application of this result in the context of linear-constrained-minimum-variance (LCMV) beamformers can be found.

Comparing the expression for $\boldsymbol{w}_H$ with that of the solution in (14), one can interpret $\boldsymbol{w}_H$ to be the minimizer of $\boldsymbol{w}^T \boldsymbol{\Omega} \boldsymbol{w}$ with constraint $\boldsymbol{G}^T \boldsymbol{w} = \boldsymbol{v}_H \triangleq \boldsymbol{\Sigma}^{-1}\boldsymbol{h}$. Fortunately, this optimization problem can be computed in a distributed manner using ADMM, since the objective is separable $\boldsymbol{w}^T \boldsymbol{\Omega} \boldsymbol{w} = \sum_{n=1}^{N} \boldsymbol{w}_{\mathcal{F}_n}^T \boldsymbol{\Omega}_{\mathcal{F}_n} \boldsymbol{w}_{\mathcal{F}_n}$. The nth constraint evaluates to $\boldsymbol{g}_{\mathcal{B}_n}^T \boldsymbol{w}_{\mathcal{T}_n} = \boldsymbol{v}_{\mathcal{H},n} = h_n/\sigma_n^2$, which is also amenable for distributed computation. The following notations establish the equivalence between Proposition 4 and our problem: $\boldsymbol{p}_n = \boldsymbol{w}_{\mathcal{F}_n}$, $F_n(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Omega}_{\mathcal{F}_n} \boldsymbol{x}$, $\boldsymbol{e}_j$ is jth column of $\boldsymbol{G}$ and $s_j = \boldsymbol{v}_{H,j}$. In addition, step (13a) can be shown to have a closed form solution, as stated below.

Proposition 5: The following ADMM iterations exactly solves problem (14),

$$\begin{aligned} \boldsymbol{w}_{\mathcal{F}_n}(t+1) &= \frac{g_n}{2}\left(\boldsymbol{\Omega}_{\mathcal{F}_n} + \frac{c g_n^2}{2}\boldsymbol{I}\right)^{-1} \left(c g_n \boldsymbol{w}_{\mathcal{F}_n}(t) - c\boldsymbol{r}_{\mathcal{B}_n} - \boldsymbol{\lambda}_{\mathcal{B}_n}\right) \\ \lambda_n(t+1) &= \lambda_n(t) + c\boldsymbol{r}_n(t+1), \ \forall n \qquad (15) \\ \boldsymbol{r}_n(t) &= \frac{1}{|\mathcal{B}_n|}\left(\boldsymbol{g}_{\mathcal{B}_n}^T \boldsymbol{w}_{\mathcal{T}_n} - \boldsymbol{v}_{\mathcal{H},n}\right), \ \forall n, \end{aligned}$$

thereby enabling decentralized computation of $\boldsymbol{w}_H$.

The proof of Proposition 5 is relegated to a future extended version of this paper. The constant c in our implementations was fixed at 1.0. For modest accuracy (residual error $\boldsymbol{r}_n(t)$ of the order 10^{-4}), the ADMM converged very fast within tens of iterations. Such fast convergence characteristic of ADMM is well known [12].

After $\boldsymbol{w}_H = (\boldsymbol{w}_{H,\mathcal{F}_1}^T, \cdots, \boldsymbol{w}_{H,\mathcal{F}_N}^T)$ is computed, the scaling parameter κ_H can be determined using consensus in order to ensure that the cumulative energy is $\mathcal{E}$. Figure 4 illustrates all the steps.

F. Universal strategy

The universal strategy linearly interpolates $\boldsymbol{w}_L$ and $\boldsymbol{w}_H$ as per Proposition 2. Having obtained the two strategies already, it remains to compute α_U in a distributed manner – which in turn depends on quantities J_0 and c_1, c_2, c_3. First, $J_0 = \boldsymbol{h}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{h}$ is computed by performing consensus on $\frac{h_n^2}{\sigma_n^2}$. In order to compute c_1 and c_2, note that $\boldsymbol{v}_L \triangleq \boldsymbol{\Gamma}^{-1}\boldsymbol{h} = \boldsymbol{G}^T \boldsymbol{w}_L$ can be computed from $\boldsymbol{w}_L$ using local interactions. Thereafter, consensus on $h_n \boldsymbol{v}_{L,n}$ yields c_1 and consensus on $\sigma_n^2 \boldsymbol{v}_{L,n}^2$ yields c_2. The quantity $c_3 = \boldsymbol{w}_H^T \boldsymbol{\Omega} \boldsymbol{w}_H$ can be computed by performing a consensus on $\boldsymbol{w}_{H,\mathcal{F}_n}^T \boldsymbol{\Omega}_{\mathcal{F}_n} \boldsymbol{w}_{H,\mathcal{F}_n}$ once the high-energy strategy is computed. Finally, after $\boldsymbol{w}_U$ is computed, the scale factor κ_U can be determined as in earlier cases. Figure 4 illustrates the various steps described above.

IV. Conclusion

In this paper, we developed a decentralized framework for distributed estimation with spatial collaboration. The motivation was to reduce the control overhead of our earlier approach [6] by introducing decentralized computation at the sensor nodes. We introduced three strategies, two simple strategies that work very well near each boundary of the entire distortion regime and a third strategy was formulated by interpolating the previous two strategies. Numerical simulations illustrated the efficiency of all the strategies. Theoretical efficiency bounds for all these strategies remain an interesting topic for future research. Also, the cost of decentralized computations will be investigated in the future.

REFERENCES

[1] S. Kar and P. K. Varshney, "Linear coherent estimation with spatial collaboration," *Information Theory, IEEE Transactions on*, vol. 59, no. 6, pp. 3532–3553, 2013.

[2] S. Cui, J.-J. Xiao, A. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Estimation diversity and energy efficiency in distributed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 9, pp. 4683–4695, Sept. 2007.

[3] A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks-Part I: Gaussian case," *Signal Processing, IEEE Transactions on*, vol. 54, no. 3, pp. 1131–1143, 2006.

[4] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. Goldsmith, "Linear coherent decentralized estimation," *Signal Processing, IEEE Transactions on*, vol. 56, no. 2, pp. 757–770, Feb. 2008.

[5] J. Fang and H. Li, "Power constrained distributed estimation with cluster-based sensor collaboration," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 7, pp. 3822–3832, July 2009.

[6] M. Gastpar, B. Rimoldi, and M. Vetterli, "To code, or not to code: Lossy source-channel communication revisited," *Information Theory, IEEE Transactions on*, vol. 49, no. 5, pp. 1147–1158, May 2003.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[8] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[9] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[10] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[11] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing*. John Wiley & Sons, 2002.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.