Information-Driven Distributed Maximum Likelihood Estimation Based on Gauss-Newton Method in Wireless Sensor Networks

Tong Zhao, *Member, IEEE*, and Arye Nehorai, *Fellow, IEEE*

***Abstract*—In this paper, we develop an energy-efficient distributed estimation method that can be used in applications such as the estimation of a diffusive source and the localization and tracking of an acoustic target in wireless sensor networks. We first propose a statistical measurement model in which we separate the linear and nonlinear parameters. This modeling strategy reduce the processing complexity. We then study the distributed implementation of the Gauss-Newton method in the maximum likelihood estimation. After that we propose a fully distributed estimation approach based on an incremental realization of the Gauss-Newton method. We derive three modifications of the basic algorithm to improve the distributed processing performance while still considering the energy restriction. We implement the idea of information-driven collaborative signal processing and provide a sensor-node scheduling scheme in which the Cramér–Rao bound (CRB) is used as the performance and information utility measure to select the next sensor node. Numerical examples are used to study the performance of the distributed estimation, and we show that of the methods considered here, the proposed multiple iteration Kalman filtering method has the most advantages for wireless sensor networks.**

***Index Terms*—Energy-efficient distributed estimation, Gauss–Newton method, maximum-likelihood estimation, sensor node scheduling, wireless sensor networks.**

I. Introduction

RECENTLY, wireless sensor networks have attracted extensive research interest as a new approach to providing clever interaction with the physical world [1]–[4]. In this paper, we address the issue of developing an energy-efficient distributed parameter estimation method for applications in wireless sensor networks. The proposed methods can be applied in many civilian and military applications, such as the estimation of a diffusive source [5]–[7], the localization and tracking of an acoustic target [8], [13], and most other passive sensing applications.

Manuscript received March 28, 2006; revised January 28, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Venkatesh Saligrama. This work was supported in part by the Department of Defense under Air Force Office of Scientific Research MURI Grant FA9550-05-1-0443, AFOSR Grant FA9550-05-1-0018, and by the National Science Foundation under Grant CCR-0330342.

The authors are with the Department of Electrical and Systems Engineering, Washington University, St. Louis, MO 63130 USA (e-mail: tzhou3@ese.wustl.edu; nehorai@ese.wustl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.896267

In a typical wireless sensor network, each sensor node operates unattended with limited battery power and communicates with others through wireless links that consume the dominant part of the energy in the network [4]. Therefore, a challenge in wireless sensor network research is to develop energy-efficient processing methods that can decrease the power usage while still maintaining a required processing performance, thus increasing the lifetime of the whole sensor network. In traditional centralized processing methods, the data measured at each sensor are transmitted to a fusion center for processing. However, even though centralized methods can achieve high processing performance, some of their inherent properties, such as the large communication energy consumption and the lack of adaptability to link failures and dynamical changes in the network topology, prohibit their use in a wireless sensor network. Hence, recent research has focused on developing energy-efficient distributed processing technologies suitable for the applications in wireless sensor networks.

Most of the current distributed estimation methods fall into two categories. In one category, the distributed estimation is developed using the sequential (incremental) Bayesian method [9]–[14]. With this method, a state belief (posterior density function) is propagated in the sensor network according to an information theoretic criterion. A typical method is the distributed particle filter [12]. A difficulty in this type of methods is that the convergence of the estimation is not easy to be proven. In another category, the distributed estimation is obtained by implementing the ordinary centralized estimation methods, such as the maximum likelihood estimation and the least-squares method in a distributed fashion [15]–[19]. These type of methods maintains the optimality of the centralized methods, whereas the required communication amount is large. In this paper, we develop a new framework for the distributed data exploration in wireless sensor networks, in which we join the advantages of the above two types of methods.

The recent relevant work is presented in [15] and [16]. In [15], a distributed Gauss-Newton method is derived. However, the difference from our work is that: i) the goal of this method is for sensor node self-localization; and ii) all sensor nodes are involved in data communications in each iteration. In [16], a space-time diffusion scheme is derived for least-square parameter estimation using peer-to-peer communication. This method relies on the information exchange among all sensor nodes during a long sampling times. The convergence is proven under a linear measurement model assumption. However, in this paper, the essential point is that under an framework of maximum likelihood estimation, we apply a information utility

measure to choose an optimal sequence of sensor node to collect measurements such that we reduce the required data communications.

More specifically, we derive our new methods by combining the distributed implementation of a Gauss-Newton-based maximum likelihood estimation and the information-driven sensor-node scheduling. We first modify the ordinary implementation of the Gauss-Newton method such that an update of the estimate in each Gauss-Newton iteration can be divided into a sequence of updates at each sensor node (we do not need to wait for the collection of data from all the sensor nodes). A performance criterion is used to determine when the estimation process will stop. Then we implement the idea of information-driven collaborative signal processing [20], i.e., we use the sensor-node scheduling to optimally allocate the limited resources such as the energy and bandwidth in a wireless sensor network. Specifically, according to a certain information measure, we find an optimal subset and/or a sensor-node querying route to process the estimation such that we can decrease energy consumption and response time while maintaining an acceptable processing performance.

We derive the information measure based on the Fisher information matrix. We find that in our distributed estimation method, calculation of the Fisher information matrixes is an intrinsic part of the algorithm, hence they can be obtained without any extra computation and data transmission, which is suitable for the application in wireless sensor networks. Another advantage of our new method is that we also take into consideration the limited processing capability of each sensor node in wireless sensor networks. For example, we propose a statistical measurement model in which we separate the linear parameters from the overall parameters such that estimation computation can be substantially decreased.

The paper is organized as follows. In Section II, we present the statistical measurement model and use the diffusive source as an example. In Section III, we derive the standard maximum likelihood estimation and analyze the distributed implementation of the Gauss-Newton method. In Section IV, we propose our new fully distributed estimation method and the sensor-node scheduling scheme. Numerical examples are used to demonstrate the performances of the proposed methods in Section V. Conclusions and ideas for future work are presented in Section VI.

II. Statistical Measurement Model

In this section, we propose a statistical measurement model based on which we derive our distributed maximum likelihood (ML) estimation method. In this model, we assume the parameters are stationary. The model can also be extended to the dynamic parameter cases. This measurement model is suitable for a broad range of applications in wireless sensor networks. We present an example of monitoring a diffusive source using wireless sensor networks.

To model the measurements, we suppose that a spatially distributed wireless sensor network has been deployed. Each sensor node in this network is located at a known position and can take measurements at a sequence of time instants. We denote the sensor node locations at the positions $\{\boldsymbol{r}_i, i = 1, 2, \ldots\}$; the sensor node i takes the measurements at time samples $\{t_j^{(i)}, j = 1, 2, \ldots, N_i\}$. Then we consider the following statistical measurement model at each sensor node:

$$y\left(\boldsymbol{r}_i, t_j^{(i)}\right) = \boldsymbol{a}^{\mathrm{T}}\left(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)}\right)\boldsymbol{x} + e\left(\boldsymbol{r}_i, t_j^{(i)}\right) \tag{1}$$

where

- $y(\boldsymbol{r}_i, t_j^{(i)})$ is the measurement of sensor node i at time $t_j^{(i)}$;
- $\boldsymbol{a}(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)}) = [a_1(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)}), \ldots, a_K(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)})]^{\mathrm{T}}$ is a K-dimensional vector, related to the underlying physical mechanism of the source we want to estimate, where $a_k(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)}), 1 \le k \le K$, is a real scalar function depending (generally nonlinearly) on $\boldsymbol{\theta}$;
- $\boldsymbol{\theta}$ is a $P \times 1$ parameter vector related to the source we are interested in; its relation to the observation $y(\boldsymbol{r}_i, t_j^{(i)})$ through $a_k(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)})$ is nonlinear;
- $\boldsymbol{x}$ is a $K \times 1$ parameter vector representing the linear part of the model (1);
- $e(\boldsymbol{r}_i, t_j^{(i)})$ represents additive noise. We assume it is a real random variable with zero mean and Gaussian distribution. We also assume that this additive noise is correlated in the time domain but independent in the space domain, i.e.,

$$\mathrm{E}\left[e\left(\boldsymbol{r}_{i_1}, t_{j_1}^{(i_1)}\right) e\left(\boldsymbol{r}_{i_2}, t_{j_2}^{(i_2)}\right)\right] = \begin{cases} \sigma_{j_1 j_2}^{(i_1)}, & i_1 = i_2 \\ 0, & i_1 \neq i_2. \end{cases}$$

In this measurement model, the term $\boldsymbol{a}(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j^{(i)})\boldsymbol{x}$ represents the underlying physical mechanism. Both parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are related to the unknown source and the environment and may include the parameters we are interested in. Note that in our model, instead of using a general nonlinear function to describe the relationship between the unknown parameters and the measurements, we represent the signal into a regression form that is a linear combination of certain regression basis functions. By providing this structure to the signal, we reduce the computation complexity of the corresponding distributed estimation methods. This is because i) the unknown parameters in $\boldsymbol{x}$ can be estimated using closed-form formulas; and ii) the nuisance parameters in $\boldsymbol{x}$ can be removed from the unknown parameter list by applying a subspace projection method.

To simplify the presentation, we assume that each sensor node takes measurements at the same sequence of time instants $\{t_j, j = 1, \ldots, N\}$ (this assumption can be relaxed without significant changes to the derived methods). We also denote $y_{ij} = y(\boldsymbol{r}_i, t_j)$, $e_{ij} = e(\boldsymbol{r}_i, t_j)$, and $\boldsymbol{a}_{ij}(\boldsymbol{\theta}) = \boldsymbol{a}(\boldsymbol{\theta}; \boldsymbol{r}_i, t_j)$. Then we can rewrite the measurement model (1) as

$$y_{ij} = \boldsymbol{a}_{ij}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{x} + e_{ij}. \tag{2}$$

When we lump all the measurements taken at sensor node i into vector form, we obtain

$$\boldsymbol{y}_i = A_i(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{e}_i, \quad \boldsymbol{e}_i \sim \mathcal{N}_N(0, \Sigma_i) \tag{3}$$

where

- $\boldsymbol{y}_i = [y_{i,1}, \ldots, y_{i,N}]^{\mathrm{T}}$, which is an $N \times 1$ vector;
- $A_i(\boldsymbol{\theta}) = [\boldsymbol{a}_{i,1}(\boldsymbol{\theta}), \ldots, \boldsymbol{a}_{i,N}(\boldsymbol{\theta})]^{\mathrm{T}}$, which is an $N \times K$ matrix;

- $\boldsymbol{e}_i = [e_{i,1}, \ldots, e_{i,\mathrm{N}}]^{\mathrm{T}}$ which is an N-dimension Gaussian distributed random vector with mean 0 and covariance matrix Σ_i. It is also independent in the space domain, i.e., $\mathrm{E}[\boldsymbol{e}_{i_1}\boldsymbol{e}_{i_2}^{\mathrm{T}}] = 0$ when $i_1 \neq i_2$.

This statistical measurement model can be used in many applications in wireless sensor networks, e.g., the monitoring of a diffusive source, the localization and tracking of an acoustic or electromagnetic target, and most other passive sensing applications. In the following, we show an example of the proposed model in the application of estimating a diffusive source which is important in the homeland security applications and environment monitoring.

A. Measurement Model of a Diffusive Source

We assume that a continuous point diffusive source is located at position $\boldsymbol{r}_0 = [x_0, y_0, z_0]$ and that it releases a diffusive substance at time t_{I}. A wireless sensor network is deployed to monitor this diffusive source. The measurement taking by sensor node i at time t_j can be written as

$$\begin{aligned} y(\boldsymbol{r}_i, t_j) &= c(\boldsymbol{r}_i, t_j) + b + e(\boldsymbol{r}_i, t_j) \\ e(\boldsymbol{r}_i, t_j) &\sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{4}$$

where $c(\boldsymbol{r}_i, t_j)$ is the concentration of the substance of interest at position $\boldsymbol{r}_i$ and time t_j. The term $c(\boldsymbol{r}_i, t_j)$ represents the underlying physical mechanism of substance dispersion, and it can be obtained by solving diffusion equations under certain boundary and initial conditions. If we assume the environment to be a semi-infinite medium with an impermeable boundary, and the constant substance release rate from the source is μ, we obtain the concentration distribution

$$\begin{aligned} c(\boldsymbol{r}, t) = \mu \int_{t_{\mathrm{I}}}^{t} & \frac{1}{8[\pi\kappa(t-\tau)]^{3/2}} \\ & \times \left\{ \exp\left[-\frac{|\boldsymbol{r} - \boldsymbol{r}_0|^2}{4\kappa(t-\tau)} \right] + \exp\left[-\frac{|\boldsymbol{r} - \boldsymbol{r}_1|^2}{4\kappa(t-\tau)} \right] \right\} \mathrm{d}\tau \end{aligned} \tag{5}$$

where $\boldsymbol{r}_1 = [x_0, y_0, -z_0]^{\mathrm{T}}$ and κ is the medium diffusivity [21]. The other items in (4) are defined as follows (see [22] for details):

- b is a bias term, representing the sensor's response to foreign substances, which is assumed to be an unknown constant;
- $e(\boldsymbol{r}_i, t_j)$ is the sensor's noise, assumed to be Gaussian distributed, independent in time and space.

If we denote $y_{ij} = y(\boldsymbol{r}_i, t_j), e_{ij} = e(\boldsymbol{r}_i, t_j)$, and $\mu\alpha_{ij}(\boldsymbol{\theta}) = c(\boldsymbol{r}_i, t_j)$, we can rewrite (4) as

$$y_{ij} = \boldsymbol{a}_{ij}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{x} + e_{ij} \tag{6}$$

which is an instance of the measurement model in the (2). Here

- $\boldsymbol{a}_{ij}(\boldsymbol{\theta}) = [\alpha_{ij}(\boldsymbol{\theta}),\ 1]^{\mathrm{T}}$;
- $\boldsymbol{x} = [\mu,\ b]^{\mathrm{T}}$ is the linear parameter vector, in which μ is the parameter of interest and b is the nuisance parameter;
- $\boldsymbol{\theta} = [x_0, y_0, z_0, t_{\mathrm{I}}, \kappa]^{\mathrm{T}}$ represents the nonlinear source and medium parameters.

III. Standard Maximum Likelihood Estimation

In this section, we first derive the standard maximum likelihood estimation for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{x}$ in the proposed measurement model (3); then we study the realization of this ML estimation using an iterative Gauss-Newton method and thereby develop a distributed implementation scheme motivated by the idea of distributed EM algorithms in [19]. Even though this distributed implementation can be used in wireless sensor networks, some important limitations of this method restrict its applications. This restriction motivates us to design a new framework for the distributed ML estimation.

A. Maximum Likelihood Estimation

We assume that M sensor nodes are activated to process an estimation task in a wireless sensor network. At sensor node i, measurement $\boldsymbol{y}_i$ follows the proposed measurement model (3). Here we assume the covariance matrix Σ_i is known, since in practical applications this matrix can be estimated during the calibration step.

We can collect the measurements $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$ from all M sensor nodes into vector form as

$$\boldsymbol{y} = A(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{e} \tag{7}$$

where

- $\boldsymbol{y} = [\boldsymbol{y}_1^{\mathrm{T}}, \ldots, \boldsymbol{y}_M^{\mathrm{T}}]^{\mathrm{T}}$, which is an $MN \times 1$ vector;
- $A(\boldsymbol{\theta}) = [A_1^{\mathrm{T}}(\boldsymbol{\theta}), \ldots, A_M^{\mathrm{T}}(\boldsymbol{\theta})]^{\mathrm{T}}$, which is an $MN \times K$ matrix;
- $\boldsymbol{e} = [\boldsymbol{e}_1^{\mathrm{T}}, \ldots, \boldsymbol{e}_M^{\mathrm{T}}]^{\mathrm{T}}$ which represents the additive Gaussian noise. According to our assumptions, it follows an MN-dimension multiple normal distribution with mean 0 and covariance matrix Σ, i.e., $\boldsymbol{e} \sim \mathcal{N}_{MN}(0, \Sigma)$ with $\Sigma = \mathrm{diag}\{\Sigma_1, \ldots, \Sigma_M\}$. Here, since the covariance matrix Σ_i is assumed to be positive definite, Σ_i^{-1} is also positive definite and can be factored as $\Sigma_i^{-1} = D_i^{\mathrm{T}} D_i$ where D_i is an $N \times N$ invertible matrix. Accordingly, the covariance matrix Σ can also be factored as $\Sigma^{-1} = D^{\mathrm{T}} D$ where $D = \mathrm{diag}\{D_1, \ldots, D_M\}$.

In this measurement model, the variables $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are the unknown parameters to be estimated.

According to the above measurement model (7), we find that the log-likelihood function of the measurements $\boldsymbol{y}$ is

$$\begin{aligned} & L(\boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{x}) \\ & \quad = -\frac{MN}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| \\ & \qquad - \frac{1}{2}[\boldsymbol{y} - A(\boldsymbol{\theta})\boldsymbol{x}]^{\mathrm{T}}\Sigma^{-1}[\boldsymbol{y} - A(\boldsymbol{\theta})\boldsymbol{x}] \\ & \quad = -\frac{MN}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{M}\log|\Sigma_i| \\ & \qquad - \frac{1}{2}\sum_{i=1}^{M}[\boldsymbol{y}_i - A_i(\boldsymbol{\theta})\boldsymbol{x}]^{\mathrm{T}}\Sigma_i^{-1}[\boldsymbol{y}_i - A_i(\boldsymbol{\theta})\boldsymbol{x}]. \end{aligned} \tag{8}$$

Maximizing this log-likelihood function, we obtain the ML estimates of $\boldsymbol{\theta}$ and $\boldsymbol{x}$

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}}\{[\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})]^{\mathrm{T}} \\ &\quad\times\Sigma^{-1}[\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})]\} \\ &= \arg\min_{\boldsymbol{\theta}}\left\{\sum_{i=1}^{M}[\boldsymbol{y}_i-A_i(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})]^{\mathrm{T}}\Sigma_i^{-1}[\boldsymbol{y}_i-A_i(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})]\right\}\end{aligned} \tag{9}$$

$$\begin{aligned}\hat{\boldsymbol{x}}(\boldsymbol{\theta}) &= [\tilde{A}^{\mathrm{T}}(\boldsymbol{\theta})\tilde{A}(\boldsymbol{\theta})]^{-1}\tilde{A}^{\mathrm{T}}(\boldsymbol{\theta})\tilde{\boldsymbol{y}} \\ &= \left[\sum_{i=1}^{M}A_i^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_i^{-1}A_i(\boldsymbol{\theta})\right]^{-1}\sum_{i=1}^{M}A_i^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_i^{-1}\boldsymbol{y}_i\end{aligned} \tag{10}$$

where $\tilde{A}(\boldsymbol{\theta}) = DA(\boldsymbol{\theta})$. We also find that the optimization in (9) can be simplified as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\{\tilde{\boldsymbol{y}}^{\mathrm{T}}P_{\tilde{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{y}}\} \tag{11}$$

where $\tilde{\boldsymbol{y}} = D\boldsymbol{y}$ and $P_{\tilde{A}}(\boldsymbol{\theta})$ is the projecting matrix on the column space of $\tilde{A}(\boldsymbol{\theta})$

$$P_{\tilde{A}}(\boldsymbol{\theta}) = \tilde{A}(\boldsymbol{\theta})[\tilde{A}^{\mathrm{T}}(\boldsymbol{\theta})\tilde{A}(\boldsymbol{\theta})]^{-1}\tilde{A}^{\mathrm{T}}(\boldsymbol{\theta}).$$

1) Nuisance Parameters: If there exist nuisance parameters in the linear parameter vector $\boldsymbol{x}$, the ML estimates $\hat{\boldsymbol{x}}(\boldsymbol{\theta})$ in (10) can be simplified. Assume the vector $\boldsymbol{x}$ is partitioned into two parts

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}} & \boldsymbol{x}_2^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$$

where $\boldsymbol{x}_1$ includes K_1 parameters of interest, $\boldsymbol{x}_2$ includes K_2 nuisance parameters, and $K = K_1 + K_2$. Correspondingly, matrix $A(\boldsymbol{\theta})$ can also be partitioned into two parts

$$A(\boldsymbol{\theta}) = \begin{bmatrix}A^{(1)}(\boldsymbol{\theta}), & A^{(2)}(\boldsymbol{\theta})\end{bmatrix}$$

where $A^{(1)}(\boldsymbol{\theta})$ includes the first K_1 columns of $A(\boldsymbol{\theta})$ while $A^{(2)}(\boldsymbol{\theta})$ includes the last K_2 columns of $A(\boldsymbol{\theta})$. If we define $\tilde{A}^{(1)}(\boldsymbol{\theta}) = DA^{(1)}(\boldsymbol{\theta})$, and $\tilde{A}^{(2)}(\boldsymbol{\theta}) = DA^{(2)}(\boldsymbol{\theta})$, we can obtain the ML estimate for the parameters of interest $\boldsymbol{x}_1$ as

$$\hat{\boldsymbol{x}}_1 = \left[\left(\breve{A}^{(1)}(\hat{\boldsymbol{\theta}})\right)^{\mathrm{T}}\breve{A}^{(1)}(\hat{\boldsymbol{\theta}})\right]^{-1}\left(\breve{A}^{(1)}(\hat{\boldsymbol{\theta}})\right)^{\mathrm{T}}\tilde{\boldsymbol{y}} \tag{12}$$

where we denote

$$\breve{A}^{(1)}(\boldsymbol{\theta}) = (I - P_{\tilde{A}^{(2)}}(\boldsymbol{\theta}))\tilde{A}^{(1)}(\boldsymbol{\theta})$$

which is the complementary projection of $\tilde{A}^{(1)}(\boldsymbol{\theta})$ on the column space of $\tilde{A}^{(2)}(\boldsymbol{\theta})$.

In a special case when the matrix $A^{(2)}(\boldsymbol{\theta})$ is not dependent on $\boldsymbol{\theta}$, i.e., $A^{(2)}$ is a constant matrix as in (6), the ML estimate for $\boldsymbol{\theta}$ in (9) can also be simplified as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\{\boldsymbol{y}^{\mathrm{T}}P_{\breve{A}^{(1)}}(\boldsymbol{\theta})\boldsymbol{y}\} \tag{13}$$

where the complexity to calculate $P_{\breve{A}^{(1)}}(\boldsymbol{\theta})$ is much less than the calculation of $P_{\tilde{A}}(\boldsymbol{\theta})$ in (9).

B. Implementation With Gauss-Newton Method

For a centralized ML estimation, the measurements from all sensors are transmitted to a fusion center, and the fusion center applies (9) and (10) to calculate the ML estimates directly. However, the severe energy restriction in wireless sensor networks prohibits the transmission of raw data from all sensor nodes to a fusion center. Therefore, the estimation results in (9) and (10) are not enough to derive a distributed estimation method. In this section, we analyze the realization of ML estimation using a Gauss-Newton method and develop a distributed implementation scheme. In this distributed implementation, in stead of transmitting all the measurements to a center for processing, only several sufficient statistics are transmitted between nearby sensor nodes; hence, we lower the communication cost.

1) Gauss-Newton Method: To obtain the ML estimates, we observe that the nonlinear optimization problem in (9) to calculate $\hat{\boldsymbol{\theta}}$ is a nonlinear weighted least squares (LS) problem (see [23] and [24]) under the assumption that the covariance matrix Σ is known. Then we can solve it using the effective and efficient *Gauss-Newton* method. Note that an iterative method such as Gauss-Newton method can only converge to a local maximum value. In general this method does not yield the maximum-likelihood estimate unless the optimization problem can be formulated as, e.g., a convex optimization problem. Therefore, the choice of the initial value for this iterative method is very important.

When employing the Gauss-Newton method to solve the nonlinear weighted LS problem in (9), we obtain an iterative process to update the estimate of $\boldsymbol{\theta}$ as

$$\begin{aligned}\boldsymbol{\theta}^{(k+1)} &= \boldsymbol{\theta}^{(k)} + \left[H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}H\left(\boldsymbol{\theta}^{(k)}\right)\right]^{-1} \\ &\quad\times H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}\left(\boldsymbol{y}-A\left(\boldsymbol{\theta}^{(k)}\right)\hat{\boldsymbol{x}}\left(\boldsymbol{\theta}^{(k)}\right)\right)\end{aligned} \tag{14}$$

where

$$H(\boldsymbol{\theta}) = \frac{\partial[A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})]}{\partial\boldsymbol{\theta}^{\mathrm{T}}} \tag{15}$$

is an $MN \times P$ Jacobian matrix. In order to obtain a distributed estimation, we study the details of the implementation in (14). First, we can partition $H(\boldsymbol{\theta})$ as

$$H(\boldsymbol{\theta}) = [\boldsymbol{h}_1(\boldsymbol{\theta}), \ldots, \boldsymbol{h}_P(\boldsymbol{\theta})] \tag{16}$$

where $\boldsymbol{h}_p(\boldsymbol{\theta})$ is the pth column of $H(\boldsymbol{\theta})$ and can be derived as

$$\begin{aligned}\boldsymbol{h}_p(\boldsymbol{\theta}) &= \frac{\partial\{A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})\}}{\partial\theta_p} \\ &= \frac{\partial\{A(\boldsymbol{\theta})\left[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})\right]^{-1}A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y}\}}{\partial\theta_p} \\ &= \frac{\partial A(\boldsymbol{\theta})}{\partial\theta_p}[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y} \\ &\quad + A(\boldsymbol{\theta})\frac{\partial\{[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}\}}{\partial\theta_p}A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y} \\ &\quad + A(\boldsymbol{\theta})[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}\frac{\partial A^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\Sigma^{-1}\boldsymbol{y} \\ &\qquad\qquad p = 1, \ldots, P.\end{aligned} \tag{17}$$

Hence, the second term on the right-hand side (RHS) of (14) can be rewritten as

$$\begin{aligned}&[H^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}H(\boldsymbol{\theta})]^{-1}H^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}(\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta}))\\&\quad=\begin{bmatrix}\boldsymbol{h}_1^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{h}_1(\boldsymbol{\theta}) & \dots & \boldsymbol{h}_1^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{h}_P(\boldsymbol{\theta})\\ \vdots & \ddots & \vdots\\ \boldsymbol{h}_P^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{h}_1(\boldsymbol{\theta}) & \dots & \boldsymbol{h}_P^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{h}_P(\boldsymbol{\theta})\end{bmatrix}^{-1}\\&\qquad\times\begin{bmatrix}\boldsymbol{h}_1^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\left(\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})\right)\\ \vdots\\ \boldsymbol{h}_P^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\left(\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})\right)\end{bmatrix}.\end{aligned}\tag{18}$$

In this formula, the term $\boldsymbol{h}_p^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\left(\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta})\right)$ can be further derived as

$$\begin{aligned}&\boldsymbol{h}_p^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}(\boldsymbol{y}-A(\boldsymbol{\theta})\hat{\boldsymbol{x}}(\boldsymbol{\theta}))\\&\quad=[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y}]^{\mathrm{T}}[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}\\&\qquad\times\frac{\partial A^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\Sigma^{-1}\boldsymbol{y}\\&\qquad-[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y}]^{\mathrm{T}}[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}\\&\qquad\times\frac{\partial A^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\Sigma^{-1}A(\boldsymbol{\theta})[A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}A(\boldsymbol{\theta})]^{-1}\\&\qquad\times A^{\mathrm{T}}(\boldsymbol{\theta})\Sigma^{-1}\boldsymbol{y}\end{aligned}\tag{19}$$

where $\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}$.

From the (17) and (19), we make the following two observations.

1) In order to update $\boldsymbol{\theta}^{(k)}$ using (14), we need only $P+1$ statistics. Each statistic is a $P\times 1$ vector, denoted by $\{\boldsymbol{w}_0,\boldsymbol{w}_1,\dots,\boldsymbol{w}_{\mathrm{P}}\}$, as follows:

$$\boldsymbol{w}_0=A^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}\boldsymbol{y}\tag{20}$$

$$\boldsymbol{w}_p=\left.\frac{\partial A^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}}\Sigma^{-1}\boldsymbol{y}\qquad p=1,\dots,P.\tag{21}$$

Therefore, for the estimation we do not need to collect the actual raw data $\boldsymbol{y}$ but only these statistics.

2) We can rewrite $\boldsymbol{w}_0$ and $\boldsymbol{w}_p, p=1,\dots,P$ as

$$\boldsymbol{w}_0=\sum_{i=1}^{M}A_i^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma_i^{-1}\boldsymbol{y}_i\tag{22}$$

$$\boldsymbol{w}_p=\sum_{i=1}^{M}\left.\frac{\partial A_i^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}}\Sigma_i^{-1}\boldsymbol{y}_i\quad p=1,\dots,P\tag{23}$$

which means that each of these statistics is a summation of the local statistics at each sensor node.

From these two observations, we obtain a distributed implementation of the ML estimation of $\boldsymbol{\theta}$, motivated by the idea of distributed EM algorithms in [19], as follows.

2) Distributed Implementation: An initial value of $\boldsymbol{\theta}$, denoted as $\boldsymbol{\theta}^{(0)}$, is provided at the beginning. Assume that at iteration step $k+1$, node $i=1$ has the current parameter estimate $\boldsymbol{\theta}^{(k)}$. The next ML estimate $\boldsymbol{\theta}^{(k+1)}$ can be computed by performing one message cycle through all the nodes. To begin, initialize the sufficient statistic $\{\boldsymbol{w}_0^{(k)},\boldsymbol{w}_1^{(k)},\dots,\boldsymbol{w}_P^{(k)}\}$ to zero. When the messages are transmitted to the sensor node i, the node performs the following two steps.

Step 1: It computes the local sufficient statistics using

$$\boldsymbol{w}_{0i}^{(k)}=A_i^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma_i^{-1}\boldsymbol{y}_i\tag{24}$$

$$\boldsymbol{w}_{pi}^{(k)}=\left.\frac{\partial A_i^{\mathrm{T}}(\boldsymbol{\theta})}{\partial\theta_p}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}}\Sigma_i^{-1}\boldsymbol{y}_i\quad p=1,\dots,P.\tag{25}$$

Note that the statistics are computed using only $\boldsymbol{\theta}^{(k)}$ and the local measurements $\boldsymbol{y}_i$; hence, these calculations can be implemented completely locally.

Step 2: It then increments the global sufficient statistics according to

$$\boldsymbol{w}_0^{(k)}:=\boldsymbol{w}_0^{(k)}+\boldsymbol{w}_{0i}^{(k)}\tag{26}$$

$$\boldsymbol{w}_p^{(k)}:=\boldsymbol{w}_p^{(k)}+\boldsymbol{w}_{pi}^{(k)},\quad p=1,\dots,P.\tag{27}$$

After that the current node i passes $\{\boldsymbol{w}_0^{(k)},\boldsymbol{w}_1^{(k)},\dots,\boldsymbol{w}_P^{(k)}\}$ and $\boldsymbol{\theta}^{(k)}$ to the node $i+1$.

At the last sensor node $i=M$, the node obtains the complete sufficient statistics $\{\boldsymbol{w}_0^{(k)},\boldsymbol{w}_1^{(k)},\dots,\boldsymbol{w}_P^{(k)}\}$ and $\boldsymbol{\theta}^{(k)}$, and uses them to calculate $\boldsymbol{\theta}^{(k+1)}$ by applying (14). We observe that for each iteration, $(P+2)P$ real valued numbers are transmitted between two sensor nodes. Hence, the transmission volume is independent of the number of measurements taken by each sensor node.

3) Discussion: Compared with the centralized ML estimation method, this distributed implementation algorithm avoids the necessity of transmitting raw data to the fusion center (only several sufficient statistics are transmitted between the nearby sensor nodes); hence, the algorithm decreases the required communication burden. However, several inherent properties of this algorithm limit its application in wireless sensor networks.

- Since the Gauss-Newton is an iterative algorithm, in order to obtain the ML estimate, the messages should cycle though all the sensor nodes several times (each cycle corresponding to one iteration). Even though the volume of communication for one cycle is not large, the potential total communication burden is still heavy, especially in cases where the algorithm converges very slowly.
- In order to update the estimate once, the messages need to cycle through all the sensor nodes, which is time costly and not suitable for some applications where a rapid response is required.
- A predetermined message transmission route is needed to guarantee that the messages pass through all the sensor nodes. This requirement is not suitable for applications in wireless sensor networks, because the network topology changes dynamically due to the failure of some sensor nodes or communication links.
- Not all the sensor nodes provide useful information to improve the estimation; furthermore, some information may be redundant. Therefore, collecting information from all available sensor nodes is not an optimal strategy for the distributed processing due to the power restriction.

In the following, we propose a new and fully distributed estimation method motivated by the above distributed implementation and the analysis. The new method addresses these limitations and is suitable for a broader range of applications in wireless sensor networks.

IV. Information-Driven Distributed Maximum Likelihood Estimation

In this section, we develop an information-driven distributed maximum likelihood estimation method based on the aforementioned Gauss-Newton implementation. We first derive the basic distributed estimation algorithm where we assume the linear parameter $\boldsymbol{x}$ is known; then we propose several modifications of the basic algorithm to improve the estimation performance. Finally, we extend the algorithm to the case when $\boldsymbol{x}$ is unknown. We introduce information-driven collaborative signal processing techniques and propose a sensor-node scheduling scheme.

A. Information-Driven Distributed Estimation Algorithm

For the case in which linear parameter $\boldsymbol{x}$ is known, the ML estimation of parameter $\boldsymbol{\theta}$ in (9) becomes the following nonlinear weighted least-squares problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{[\boldsymbol{y} - A(\boldsymbol{\theta})\boldsymbol{x}]^{\mathrm{T}} \Sigma^{-1} [\boldsymbol{y} - A(\boldsymbol{\theta})\boldsymbol{x}]\} \tag{28}$$

which is equivalent to (9) except that $\hat{\boldsymbol{x}}(\boldsymbol{\theta})$ is replaced by a constant vector $\boldsymbol{x}$. Then, the corresponding Gauss-Newton iteration is

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + [H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right) \Sigma^{-1} H\left(\boldsymbol{\theta}^{(k)}\right)]^{-1} \times H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right) \Sigma^{-1} (\boldsymbol{y} - A\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x}) \tag{29}$$

where

$$H(\boldsymbol{\theta}) = \frac{\partial [A(\boldsymbol{\theta})\boldsymbol{x}]}{\partial \boldsymbol{\theta}^{\mathrm{T}}}.$$

In order to develop our distributed estimation method, we introduce the following theorem, in which we prove that an update of the estimate in each iteration of the Gauss-Newton method can be divided into a sequence of updates at each sensor node. This theorem implements the Proposition 1 in [25] into the derivation of distributed estimation methods.

Theorem 4.1: We define

$$H_i(\boldsymbol{\theta}) = \frac{\partial [A_i(\boldsymbol{\theta})\boldsymbol{x}]}{\partial \boldsymbol{\theta}^{\mathrm{T}}}, \quad i = 1, \ldots, M$$

and assume that the matrix $H_1^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right) \Sigma_1^{-1} H_1\left(\boldsymbol{\theta}^{(k)}\right)$ is positive definite. Then, to solve the nonlinear LS problem (28), the kth iteration of the Gauss-Newton method in (29) can be generated by the algorithm

$$\boldsymbol{\psi}_i = \boldsymbol{\psi}_{i-1} + \Gamma_i^{-1} H_i^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right) \Sigma_i^{-1} \times \left[\boldsymbol{y}_i - A_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} + H_i\left(\boldsymbol{\theta}^{(k)}\right)(\boldsymbol{\theta}^{(k)} - \boldsymbol{\psi}_{i-1})\right], \quad i = 1, \ldots, M \tag{30}$$

where we let $\boldsymbol{\psi}_0 = \boldsymbol{\theta}^{(k)}$ and we obtain $\boldsymbol{\psi}_M = \boldsymbol{\theta}^{(k+1)}$. The positive definite matrixes Γ_i are generated by

$$\Gamma_i = \Gamma_{i-1} + H_i^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right) \Sigma_i^{-1} H_i\left(\boldsymbol{\theta}^{(k)}\right), \quad i = 1, \ldots, M \tag{31}$$

with

$$\Gamma_0 = 0.$$

Proof: See the Appendix.

In the above theorem, we replace $\boldsymbol{\theta}^{(k)}$ by $\boldsymbol{\psi}_{i-1}$ in (30). Then, the updating algorithm is modified as

$$\boldsymbol{\psi}_i = \boldsymbol{\psi}_{i-1} + \Gamma_i^{-1} H_i^{\mathrm{T}}(\boldsymbol{\psi}_{i-1}) \Sigma_i^{-1} [\boldsymbol{y}_i - A_i(\boldsymbol{\psi}_{i-1})\boldsymbol{x}] \tag{32}$$

where

$$\Gamma_i = \Gamma_{i-1} + H_i^{\mathrm{T}}(\boldsymbol{\psi}_{i-1}) \Sigma_i^{-1} H_i(\boldsymbol{\psi}_{i-1}), \quad i = 1, \ldots, M. \tag{33}$$

After we make this modification, in the implementation of the Gauss-Newton method we linearize the function $A_i(\boldsymbol{\theta})\boldsymbol{x}$ at the current estimate $\boldsymbol{\psi}_{i-1}$ rather than at the estimate $\boldsymbol{\theta}^{(k)}$ available at the start of the iteration. Therefore, the resulting update of the estimate at the end of each Gauss-Newton iteration is more close to the true value than the result without this modification.

From Theorem 4.1, we observe that when applying the Gauss-Newton method to the ML estimation, the update of the estimate in each iteration can be divided into M sequential updates $\boldsymbol{\psi}_i, i = 1, \ldots, M$. In each of these updates, only the local measurements $\boldsymbol{y}_i$ and the variable $\boldsymbol{\psi}_{i-1}$ and matrix Γ_{i-1} passed from the previous sensor nodes are needed; hence, these updates can be processed completely locally. This theorem motivates us to develop a new distributed estimation method, in which the messages are transmitted through the chosen sensor nodes, and the estimate of $\boldsymbol{\theta}$ is updated when the new data at each sensor node are obtained. Then we do not need to wait for the accumulation of the measurements from all the sensor nodes in order to obtain a new estimate.

Another advantage of the new method is that it is convenient for us to implement information-driven dynamic collaborative information processing, which is a very useful technique to be applied in wireless sensor networks. Wireless sensor networks are characterized by limited battery power, frequent node attrition, and varying data and communication quality. Hence, good collaboration among distributed sensor nodes is necessary to improve estimation performance and reduce power consumption.

In our case, it is critical to select an optimal subset and an optimal order of incorporating these measurements into our estimate update. This process is called the *sensor-node scheduling*. Usually this scheduling process provides a faster reduction in estimation uncertainty compared with blind or nearest neighbor sensor node selection schemes, and it incurs a lower communication burden for meeting a required estimation performance threshold. By applying this idea in our method, instead of transmitting the messages through all the available sensor nodes in sequence, at each sensor node we use some information measures to determine to which sensor node in the neighbor of the current node we should transmit the current messages. This estimation process will continue until the final performance fits our requirements.

According to Theorem 4.1 and the above discussion, we propose a new information-driven distributed ML estimation method as follows.

Distributed estimation algorithm:

- The distributed estimation algorithm is initialized by sensor node $i = 1$. We assume an initial value of the parameter $\boldsymbol{\theta}$ is available at this sensor node, denoted by $\hat{\boldsymbol{\theta}}^{(0)}$; we let $\Gamma_0 = 0$.
- At sensor node $i, i \geq 1$, we perform the following steps:

Step 1. *Data receiving*: the sensor node is activated by the previous node and receives the transmitted data Γ_{i-1} and $\hat{\boldsymbol{\theta}}^{(i-1)}$. Note: when $i = 1$, this step is omitted.

Step 2. *New variable calculation*: Calculate the new matrix $H_i(\hat{\boldsymbol{\theta}}^{(i-1)})$ as

$$H_i\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right) = \left.\frac{\partial[A_i(\boldsymbol{\theta})\boldsymbol{x}]}{\partial \boldsymbol{\theta}^{\mathrm{T}}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(i-1)}}; \tag{34}$$

then, update the matrix Γ_i as

$$\Gamma_i = \Gamma_{i-1} + H_i^{\mathrm{T}}\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right)\Sigma_i^{-1}H_i\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right). \tag{35}$$

Step 3. *Estimate update*: Update and obtain the current estimate $\hat{\boldsymbol{\theta}}^{(i)}$ as

$$\hat{\boldsymbol{\theta}}^{(i)} = \hat{\boldsymbol{\theta}}^{(i-1)} + \Gamma_i^{-1}H_i^{\mathrm{T}}\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right) \times\Sigma_i^{-1}\left[\boldsymbol{y}_i - A_i\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right)\boldsymbol{x}\right]. \tag{36}$$

Step 4. *Estimation quality test*: Test the quality of the updated estimate according to some performance measure, e.g., the trace or determinant of the covariance matrix. If the estimate is "good enough," the estimation process is terminated; otherwise, the algorithm continues with the following steps.

Step 5. *Sensor node selection*: Select a sensor node from its neighbor according to certain information utility criteria (the details are discussed in Section IV-D).

Step 6. *Data transmission*: The current sensor node wakes up and transmits the current estimate $\hat{\boldsymbol{\theta}}^{(i)}$ and Γ_i to the selected sensor node, and then the current node returns to sleeping status.

When comparing this new distributed ML estimation algorithm with the ordinary Gauss-Newton method, we observe two characteristics: i) the new algorithm corresponds to just one iteration of the ordinary Gauss-Newton method; therefore, we decrease the required communications and shorten the response time; and ii) we linearize the nonlinear function $A_i(\boldsymbol{\theta})\boldsymbol{x}$ at the current estimate rather than at the estimate available at the beginning of the iteration; hence, we speed the convergence of the algorithm.

It can be verified that after we make these changes, the new method has the same update steps as the extended Kalman filter (EKF) specialized to the case where the state of the underlying dynamic system stays constant [25]. Because of this relationship, we can obtain a new explanation of the proposed distributed estimation method under the framework of the sequential Bayesian estimation. We linearize each new measurement around the current estimate and treat the measurement as if it were linear. At that point the belief is approximated by a Gaussian density function, and it is updated when the new measurement is available, until it is "good enough." The mean of the final obtained belief is calculated as the estimate of the unknown parameters.

B. Modifications to Improve Performance

This algorithm provides a basic framework for the distributed estimation. However, its estimation performance is limited under the restriction of the available energy. In this section, we propose three modifications of the above basic method to improve estimation performance and reduce processing time while still maintaining a low communication energy consumption.

1) Levenberg-Marquardt Method: One important problem of the ordinary Gauss-Newton method is that in each iteration to update the estimate, even though the update is in the correct direction, the Gauss-Newton method may take steps that are too long, thereby slowing the converge speed. An efficient modification is to apply a trust region strategy [26], in which when we update the estimate, we limit the distance between the new and old estimates below a bound, i.e., $\| \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)} \| \leq \varepsilon$. The solution to this problem when applied to our measurement model is

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \left[H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}H\left(\boldsymbol{\theta}^{(k)}\right) + \delta I\right]^{-1} \times H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}\left(\boldsymbol{y} - A\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x}\right) \tag{37}$$

where δ is a positive scalar. This formulation is known as the *Levenberg-Marquardt* method. It can be verified that we can implement the *Levenberg-Marquardt* method to our new distributed estimation method by setting the initial condition $\Gamma_0 = \delta I$ instead of $\Gamma_0 = 0$, so that matrix $\Gamma_1 = \delta I + H_1^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(0)})\Sigma_1^{-1}H_1(\hat{\boldsymbol{\theta}}^{(0)})$ is positive definite even if matrix $H_1^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(0)})\Sigma_1^{-1}H_1(\hat{\boldsymbol{\theta}}^{(0)})$ is not.

2) Multiple Local Iteration Implementation: As we discussed before, the new distributed estimation method can be implemented as an extended Kalman filter. The basic idea of the extended Kalman filter is to linearize the measurement function at the current estimate so it can be treated as a linear function. In the new distributed estimation algorithm, after we update the estimate using (36) at sensor node i, the new estimate $\hat{\boldsymbol{\theta}}^{(i)}$ is probably closer to the true value of $\boldsymbol{\theta}$ than was $\hat{\boldsymbol{\theta}}^{(i-1)}$. Therefore, if we relinearize function $A(\boldsymbol{\theta})\boldsymbol{x}$ at the new estimate $\hat{\boldsymbol{\theta}}^{(i)}$ and perform the update step in (36) at the same sensor node again, we would on average reduce the approximation error and increase the estimation accuracy [27]. Hence, instead of using (36) in Step 3. *Estimate update* to obtain the new estimate, we apply the following multiple local iteration updating step:

Step 3. *Estimate update*: We denote $\boldsymbol{\phi}_0 = \hat{\boldsymbol{\theta}}^{(i-1)}$ and $\Upsilon_0 = \Gamma_{i-1}$. At the sensor node i, we update the estimate using the following multiple iteration scheme

$$\boldsymbol{\phi}_j = \boldsymbol{\phi}_{j-1} + \Upsilon_j^{-1}H_i^{\mathrm{T}}(\boldsymbol{\phi}_{j-1})\Sigma_i^{-1}\left[\boldsymbol{y}_i - A_i(\boldsymbol{\phi}_{j-1})\boldsymbol{x}\right] \tag{38}$$

where

$$\Upsilon_j = \Upsilon_{j-1} + H_i^{\mathrm{T}}(\boldsymbol{\phi}_{j-1})\Sigma_i^{-1}H_i(\boldsymbol{\phi}_{j-1}), \quad j = 1, 2, \ldots. \tag{39}$$

When this iteration converges, the values $\boldsymbol{\phi}_j$ and Υ_j are used as new estimate $\hat{\boldsymbol{\theta}}^{(i)}$ and Γ_i, respectively.

Another consideration underlying the above modification is that in wireless sensor networks, communications are the major source of power consumption, rather than data processing. Therefore, in this modification we employ more effective and intensive calculations at each sensor node and thus potentially reduce the total communication burden.

3) Iterated Kalman Filtering Implementation: In our proposed distributed estimation method, we implement a single iteration of the Gauss-Newton method, which becomes an extended Kalman filter. The consideration for this single iteration is to decrease the required communications and speed the task response. However, the estimation accuracy is degraded under this scenario. Since our new method is not absolutely a ML estimation, even though the variance of the estimation error converges to zero, the estimate itself may converge to a biased value [28].

An improvement designed to address this problem extends the proposed single iteration method to a multiple global iteration method, which becomes an iterated Kalman filter (IKF) [29]. To apply this IKF implementation into our distributed estimation method, we first carry out the single iteration distributed algorithm and select a subset of the sensor nodes; then the obtained estimate is transmitted back and cycles through the chosen sensor nodes and is updated again, until it converges.

Even though we process multiple iterations with this method, it still has advantages compared with the distributed implementation of the ordinary Gauss-Newton method in Section III-B–2.

- In the first iteration, we choose a subset of the sensor nodes that contains information closely related to our estimation task. Then the messages need to cycle through only the selected sensor nodes. Therefore, we still save much communication energy compared with the ordinary distributed implementation based on Gauss-Newton.
- Even though we use multiple iterations, an estimate update is still obtained at each sensor node, rather than at the end of the current iteration. Therefore, an estimation result can be provided at any time according to the tradeoff between the required estimation accuracy and the processing time.

C. Distributed Estimation With Unknown Linear Parameters

The derivation of the above new distributed estimation method is based on the fact that the cost function we want to optimize can be written as a summation of several local component functions; each component function is related only to the measurements at the current sensor node. However, when the linear parameter $\boldsymbol{x}$ is unknown, its ML estimate is obtained using the measurements $\{\boldsymbol{y}_i, i = 1, \ldots, M\}$ from all the sensor nodes, as in (10). Hence, the cost function, as given in (9), cannot be formulated as a summation of several local component functions. Therefore, the above proposed distributed estimation method cannot be directly applied to the cases where $\boldsymbol{x}$ is unknown.

A straightforward scheme to implement the proposed distributed estimation method under such a situation is not to separate the linear parameters from the total parameters in the measurement model (7), i.e., we include $\boldsymbol{x}$ into the unknown parameter $\boldsymbol{\theta}$. Under this situation, the above proposed distributed estimation method can be directly applied when we replace $A_i(\boldsymbol{\theta})\boldsymbol{x}$ with another function, e.g., $\boldsymbol{g}_i(\boldsymbol{\theta})$. However, in this scheme, we cannot benefit from the separation of linear and nonlinear parameters in the measurement model, so the resulting computation complexity and the volume of transmitted data are increased significantly. Therefore, we propose the following more efficient estimation scheme for the case in which $\boldsymbol{x}$ is unknown.

In this new scheme, we still separate the linear parameter $\boldsymbol{x}$ from the total parameters. However, we reformulate the ML estimation of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ in (9) and (10) as the following iterative process:

$$\hat{\boldsymbol{x}}^{(i)} = \left[\sum_{j=1}^{i} A_j^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_j^{-1}A_j(\hat{\boldsymbol{\theta}}^{(i-1)})\right]^{-1} \times \sum_{j=1}^{i} A_j^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_j^{-1}\boldsymbol{y}_j \tag{40}$$

$$\hat{\boldsymbol{\theta}}^{(i)} = \arg\min_{\boldsymbol{\theta}} \left\{\sum_{j=1}^{i} \left[\boldsymbol{y}_j - A_j(\boldsymbol{\theta})\hat{\boldsymbol{x}}^{(i)}\right]^{\mathrm{T}} \times \Sigma_j^{-1}\left[\boldsymbol{y}_j - A_j(\boldsymbol{\theta})\hat{\boldsymbol{x}}^{(i)}\right]\right\}, \quad i = 1, 2, \ldots. \tag{41}$$

This iterative process generates a sequence of estimates

$$\hat{\boldsymbol{\theta}}^{(0)} \to \hat{\boldsymbol{x}}^{(1)} \to \hat{\boldsymbol{\theta}}^{(1)} \to \hat{\boldsymbol{x}}^{(2)} \to \hat{\boldsymbol{\theta}}^{(2)} \to \cdots$$

which converges to the ML estimates of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ [24].

According to this iterative process, the proposed new distributed estimation algorithm in Section IV–A can still be applied after we introduce two new variables, a vector $\boldsymbol{w}_i$ and a matrix Ξ_i, and make the following two modifications of the distributed estimation algorithm in Section IV-A at sensor node i.

1. Modify Step 2. *New variable calculation* as follows.

   Calculate the new matrix $H_i(\hat{\boldsymbol{\theta}}^{(i-1)})$ as

   $$H_i(\hat{\boldsymbol{\theta}}^{(i-1)}) = \left.\frac{\partial[A_i(\boldsymbol{\theta})\hat{\boldsymbol{x}}^{(i-1)}]}{\partial\boldsymbol{\theta}^{\mathrm{T}}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(i-1)}}; \tag{42}$$

   then, update the matrix Γ_i as

   $$\Gamma_i = \Gamma_{i-1} + H_i^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_i^{-1}H_i(\hat{\boldsymbol{\theta}}^{(i-1)}); \tag{43}$$

   update the matrix Ξ_i as

   $$\Xi_i = \Xi_{i-1} + A_i^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_i^{-1}A_i(\hat{\boldsymbol{\theta}}^{(i-1)}); \tag{44}$$

   update the vector $\boldsymbol{w}_i$ as

   $$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + A_i^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_i^{-1}\boldsymbol{y}_i. \tag{45}$$

2. Modify Step 3. *Estimate update* as follows:

Update and obtain the current estimate $\hat{\boldsymbol{x}}^{(i)}$ and $\hat{\boldsymbol{\theta}}^{(i)}$ as

$$\hat{\boldsymbol{x}}^{(i)} = \Xi_i^{-1}\boldsymbol{w}_i \tag{46}$$

$$\hat{\boldsymbol{\theta}}^{(i)} = \hat{\boldsymbol{\theta}}^{(i-1)} + \Gamma_i^{-1}H_i^{\mathrm{T}}(\hat{\boldsymbol{\theta}}^{(i-1)})\Sigma_i^{-1} \times \left[\boldsymbol{y}_i - A_i(\hat{\boldsymbol{\theta}}^{(i-1)})\hat{\boldsymbol{x}}^{(i)}\right]. \tag{47}$$

D. Sensor-Node Scheduling

Sensor-node scheduling is an important strategy used in wireless sensor networks. Through this scheduling, we can determine the optimal scheme to allocate the limited resources available in the wireless sensor network and potentially obtain high processing performance. In our proposed distributed estimation method, sensor-node scheduling involves two tasks:

1. **Sensor node selection.** We use an information utility measure to determine which sensor node in the neighbor of the current node we should select next and include its measurements into our estimation task. This sensor node should provide the information that has the most potential to decrease the estimation uncertainty and increase the performance.
2. **Termination strategy.** We need to find a proper estimation performance criterion to determine when we should stop including the new sensor nodes into our estimation process, such that we can obtain an optimal subset from all available sensor nodes. The nodes in this subset should carry the necessary and sufficient information for our estimation task.

The neighbor of the current sensor node is defined as a subset of the sensor nodes that are located within the communication range of the current node. Note that in our sensor node selection scheme we want to choose the next node that is the most informative. However, consider a tradeoff between the communication cost and the processing performance, we can only choose the sensor nodes that are within certain radio range of the current node.

The measures for the estimation performance and the information utility include: covariance-based measures, Fisher information matrix-based measures, entropy of estimation uncertainty, volume of high probability region, sensor geometry-based measures, mutual information-based measures, and so on [4]. Different measures will be used according to the sensor measurement models, the available computation capabilities and energy resources, and the required estimation accurate and processing time. In our method, we propose to use the Cramér–Rao bound (CRB) as the estimation performance and information utility measures. The reasons are: 1) The proposed distributed estimation is based on the ML estimation which is asymptotically efficient, i.e., its variance attains the CRB asymptotically [30]. The CRB is the lower bound on the variance of any unbiased estimators; it equals the inverse of the Fisher information matrix (FIM). Hence, we calculate our measures directly from the FIM. 2) We will find that the calculation of the FIM as a performance measure is an intrinsic part of the algorithm, i.e., they can be obtained without any extra computation and data transmission. This is suitable for the applications in wireless sensor networks. 3) When we use the FIM as an information measure to select the next sensor node, the FIM is computed by averaging over all possible values of the measurements from the next sensor node; therefore, the current sensor node does not require the transmission of measurements from its neighbor and hence avoids communicating useless information. The similar idea has been used in [31] for adaptive electromagnetic-induction sensing of buried targets.

In order to compute the measures, we first derive a recursive process to update the FIM when information from the new sensor node is obtained. We denote the matrix $F_i(\boldsymbol{\theta})$ in the sequence $\{F_i(\boldsymbol{\theta}), i = 1, 2, \ldots\}$ as the FIM of the nonlinear parameter $\boldsymbol{\theta}$ when we collect the measurements from the first i sensor nodes, i.e., $\boldsymbol{y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i\}$; we have a similar notation for the FIM sequence $\{F_i(\boldsymbol{x}), i = 1, 2, \ldots\}$ for the linear parameter $\boldsymbol{x}$. According to the definition of FIM, we have

$$\begin{aligned} F_i(\boldsymbol{\theta}) &= -\mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{y};\boldsymbol{\theta},\boldsymbol{x})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right] \\ &= F_{i-1}(\boldsymbol{\theta}) - \mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{y}_i;\boldsymbol{\theta},\boldsymbol{x})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right]. \end{aligned} \tag{48}$$

The second term on the RHS of (48) can be calculated as

$$\begin{aligned} &-\mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{y}_i;\boldsymbol{\theta},\boldsymbol{x})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right] \\ &\quad= \mathrm{E}\left[\frac{\partial \ln p(\boldsymbol{y}_i;\boldsymbol{\theta},\boldsymbol{x})}{\partial\boldsymbol{\theta}}\frac{\partial \ln p(\boldsymbol{y}_i;\boldsymbol{\theta},\boldsymbol{x})}{\partial\boldsymbol{\theta}^{\mathrm{T}}}\right] \\ &\quad= H_i^{\mathrm{T}}(\boldsymbol{\theta},\boldsymbol{x})\Sigma_i^{-1}H_i(\boldsymbol{\theta},\boldsymbol{x}) \end{aligned} \tag{49}$$

where

$$H_i(\boldsymbol{\theta},\boldsymbol{x}) = \left[\frac{\partial[A_i(\boldsymbol{\theta})\boldsymbol{x}]}{\partial\theta_1} \;\cdots\; \frac{\partial[A_i(\boldsymbol{\theta})\boldsymbol{x}]}{\partial\theta_{\mathrm{P}}}\right]. \tag{50}$$

Hence, we obtain the recursive equation for the FIM sequence $\{F_i(\boldsymbol{\theta}), i = 1, 2, \ldots\}$

$$F_i(\boldsymbol{\theta}) = F_{i-1}(\boldsymbol{\theta}) + H_i^{\mathrm{T}}(\boldsymbol{\theta},\boldsymbol{x})\Sigma_i^{-1}H_i(\boldsymbol{\theta},\boldsymbol{x}) \tag{51}$$

with $F_0(\boldsymbol{\theta}) = 0$. Similarly, we can obtain the recursive equation for $\{F_i(\boldsymbol{x}), i = 1, 2, \ldots\}$ as

$$F_i(\boldsymbol{x}) = F_{i-1}(\boldsymbol{x}) + A_i^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_i^{-1}A_i(\boldsymbol{\theta}) \tag{52}$$

with $F_0(\boldsymbol{x}) = 0$. By using the recursive equations in (51) and (52), we can obtain measures for estimation performance and information utility. Here, we observe an interesting result that when we compare (51) and (52) with (43) and (44) in the proposed distributed estimation algorithm, we find that the updating formulas for the matrixes Γ_i and Ξ_i are the same as the FIM $F_i(\boldsymbol{\theta})$ and $F_i(\boldsymbol{x})$, respectively. These equivalences represent the essential significance of the matrixes Γ_i and Ξ_i used in our method. It also shows that using FIM as the performance measure will not increase the computation complexity and the required transmission.

We can directly calculate the estimation performance measure using the determinant or the trace of the FIM $F_i(\boldsymbol{\theta})$ and $F_i(\boldsymbol{x})$. At sensor node i, we first update the FIMs according to (51) and (52); then we compare their determinants or traces with a predetermined threshold: if their determinants or traces equal or exceed the threshold, we will not include further measurements from the new sensor nodes into our estimation task.

For the sensor-node selection, among the neighbors of the current sensor node, we want to choose a node that will provide the most important information to reduce the estimation uncertainty. For this purpose, we use the following recursive equations to predict the FIM when we include the measurements from the next sensor node:

$$F_{i+1}(\boldsymbol{\theta}) = F_i(\boldsymbol{\theta}) + H_{i+1}^{\mathrm{T}}(\boldsymbol{\theta},\boldsymbol{x})\Sigma_{i+1}^{-1}H_{i+1}(\boldsymbol{\theta},\boldsymbol{x}) \quad (53)$$

$$F_{i+1}(\boldsymbol{x}) = F_i(\boldsymbol{x}) + A_{i+1}^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_{i+1}^{-1}A_{i+1}(\boldsymbol{\theta}). \quad (54)$$

Since $F_i(\boldsymbol{\theta})$ and $F_i(\boldsymbol{x})$ are always the same for any sensor nodes in the neighbor of the current node i, we can use only the matrixes $H_l^{\mathrm{T}}(\boldsymbol{\theta},\boldsymbol{x})\Sigma_l^{-1}H_l(\boldsymbol{\theta},\boldsymbol{x})$ and $A_l^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_l^{-1}A_l(\boldsymbol{\theta})$ to calculate the information utility measure. Here, $l \in S$ and S is a collection of the sensor nodes that are in the neighborhood of the current sensor node i. For example, if we use the trace of the matrix (denoted as $\mathrm{Tr}\{\cdot\}$) as the information measure, we can create an information utility function:

$$I(l;\boldsymbol{\theta},\boldsymbol{x}) = \mathrm{Tr}\left\{H_l^{\mathrm{T}}(\boldsymbol{\theta},\boldsymbol{x})\Sigma_l^{-1}H_l(\boldsymbol{\theta},\boldsymbol{x})\right\} + \gamma\,\mathrm{Tr}\left\{A_l^{\mathrm{T}}(\boldsymbol{\theta})\Sigma_l^{-1}A_l(\boldsymbol{\theta})\right\} \quad (55)$$

where the parameter γ determines the tradeoff between the contributions from the FIM of $\boldsymbol{\theta}$ and $\boldsymbol{x}$. Then, we will select the sensor node l that maximizes this information utility function as the next sensor node $i+1$:

$$i+1 = \arg\max_{l\in\mathcal{S}} I(l;\boldsymbol{\theta},\boldsymbol{x}). \quad (56)$$

Since we need the values of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ to compute this information utility, we can use the current estimates $\hat{\boldsymbol{\theta}}^{(i)}$ and $\hat{\boldsymbol{x}}^{(i)}$ to replace their true values.

V. Numerical Examples

In this section we use numerical examples of localizing a diffusive source to analyze and demonstrate the performance of the proposed distributed estimation methods. The proposed basic distributed estimation and its modifications can be summarized as the following four methods.

1. *Single local iteration implementation*. This method is the proposed basic Gauss-Newton-based distributed maximum likelihood estimation method in Section IV-A. It processes a single iteration at each sensor node. Since this method is equivalent to an extended Kalman filter, we denote it as the **single iteration Kalman filtering** (SIKF).
2. *Multiple local iterations implementation*. This is a modification of the above basic distributed estimation method, as we presented in Section IV-B-2. In this algorithm, we process multiple iterations at each sensor node to improve the estimation performance. We denote this method as the **multiple iteration Kalman filtering** (MIKF).
3. *Iterated Kalman filter implementation with single local iteration*. This is another modification of the basic distributed estimation method, as we presented in Section IV-B-3. In this method, we process several iterations over all the selected sensor nodes to increase the estimation accuracy. At each sensor node, we apply a single iteration Gauss-Newton step. Hence, we denote this method as the **iterated Kalman filtering with single iteration** (IKFSI).
4. *Iterated Kalman filtering implementation with multiple local iterations*. This method is similar to the IKFSI, except that at each sensor node, we process multiple local Gauss-Newton iterations. Hence, we denote it as the **iterated Kalman filtering with multiple iterations** (IKFMI).

Now we use numerical examples to compare the performance of these four methods. Through these comparisons, we investigate the properties of the proposed distributed estimation methods, and draw a conclusion concerning which method best fits the applications in wireless sensor networks when considering both estimation performance and consumption of the network resources, particularly energy.

In these numerical examples, we use the measurement model for a diffusive source in (6). Here, source position (x_0, y_0) is the unknown we need to estimate. We apply the *Levenberg-Marquardt* method, i.e., instead of setting the initial value for Γ_i to be 0, we let $\Gamma_0 = \delta I$ where δ is chosen according to various scenarios to obtain a possible best performance. For the setup of the sensor network, we use 100 wireless sensor nodes randomly deployed in a square area of 100×100 m. We define a pair of neighboring sensor nodes whose distance is less than 20 m. For the diffusion model, we consider the environment as a homogeneous semiinfinite medium with an impermeable boundary, which can represent dispersion in air above the ground. We use a scenario of a stationary impulse source located at position $\boldsymbol{r}_0 = (50, 30, 0)$ meter. The bias term in (4) is $b = 10^{-5}$ g/m^3, and the noise standard deviation is $\sigma = 6 \times 10^{-6}$ g/m^3. We take 10 temporal samples at each sensor node with a sampling interval of 5 s. The other parameters μ, κ, and t_{I} are taken to be 1 g/s, 20 m^2/s, and 0 s, respectively.

In the first example, we compare the estimation performance of the SIKF method and the MIKF method. In Fig. 1 we show the message transmission route (sensor-node selection route) when applying the MIKF method and the proposed sensor-node scheduling scheme. We observe that even though the initial sensor node, located at $\boldsymbol{r} = (20, 18, 0)$ meter, is relatively far from the source, the messages are still transmitted to the neighbor of the source. This demonstrates the optimality of the sensor node selection algorithm: according to the dispersion mechanism, the concentration near the diffusive source is relatively high; hence, the data measured by the nodes close the source have a high signal-to-noise ratio (SNR) and provide significant contributions to improve the accuracy of the distributed estimation method.

In Figs. 2 and 3 we compare the estimation bias $\sqrt{\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|}$ and the log-determinant of the CRB matrix of $\boldsymbol{\theta}$ with respect to the number of the chosen sensor nodes for the methods SIKF and MIKF. We find that the estimation performance of MIKF is

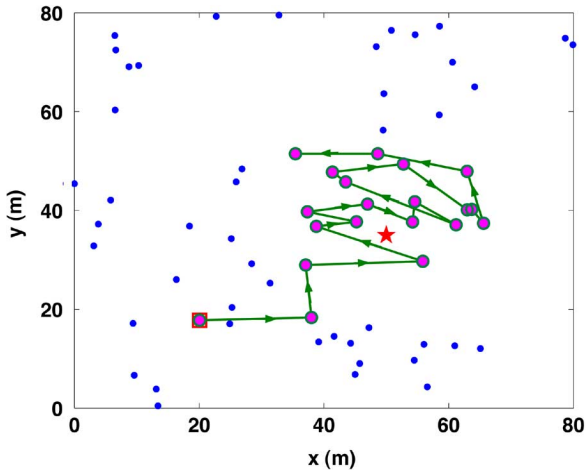


Fig. 1. Message transmission routing for localizing a diffusion source in a square area with 100 sensor nodes randomly placed. "Star" denotes the source position; "square" denotes the initial sensor node; "circle" denotes the selected sensor nodes.

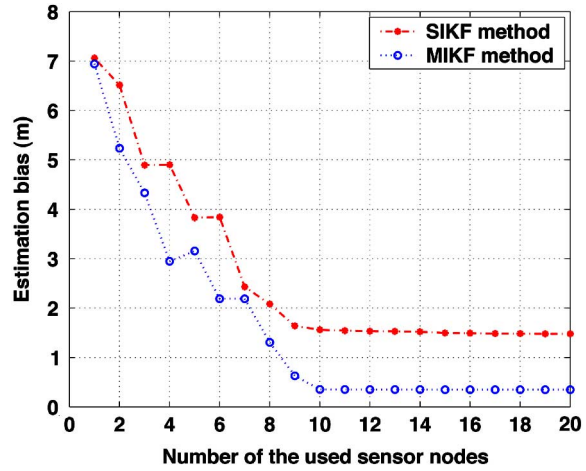


Fig. 2. Estimation bias versus number of selected sensor nodes for methods SIKF and MIKF.

much better than the SIKF, which means that multiple iterations at each sensor node can greatly improve the accuracy of the estimation without increasing the communication burden.

In the second example, we compare the performance of the SIKF with the IKFSI. Here, the initial sensor node is located at $\boldsymbol{r} = (47, 41, 0)$ meter. In the IKFSI method, the first iteration is used to select a subset of the sensor nodes that carries the most information. We set the threshold of the log-determinant of the CRB to be -4.45, and 8 sensor nodes around the source are chosen. The comparison results are shown in Figs. 4 and 5. We find that for the SIKF, the bias and the log-determinant of the CRB nearly do not decrease after 10 sensor nodes are included. This means the chosen sensor nodes $i, i > 10$ provide little information to improve the estimation performance. We also find that the performance of the IKFSI is substantially better than the SIKF even though these two methods use the same transmission, which demonstrates that global multiple iterations over all the selected sensor nodes can decrease estimation bias and improve performance. However, the cost is that the IKFSI consumes more communication energy to obtain this improvement.

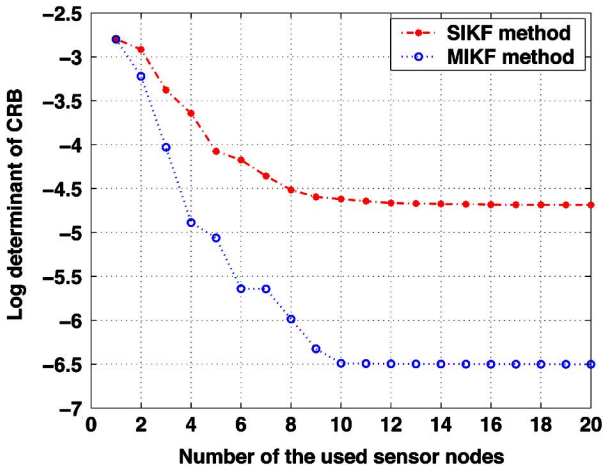


Fig. 3. Log determinant of the CRB versus number of selected sensor nodes for methods SIKF and MIKF.

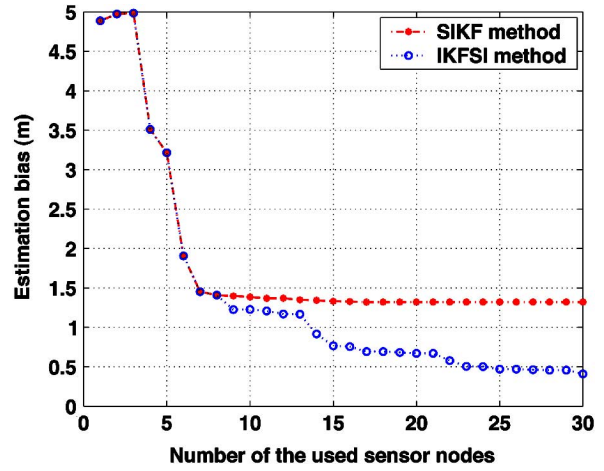


Fig. 4. Estimation bias versus number of selected sensor nodes for methods SIKF and IKFSI.

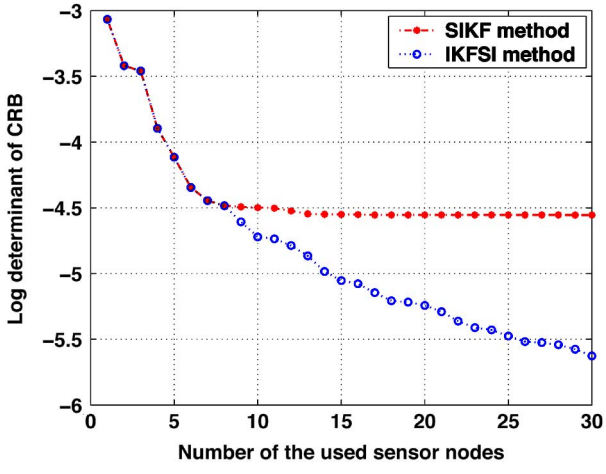


Fig. 5. Log determinant of the CRB versus number of selected sensor nodes for methods SIKF and IKFSI.

In the last example, we compare the performance of the MIKF, the IKFSI, and the IKFMI methods. The comparison results are shown in Figs. 6 and 7. We observe that the estimation accuracy and performance of the MIKF is better than the

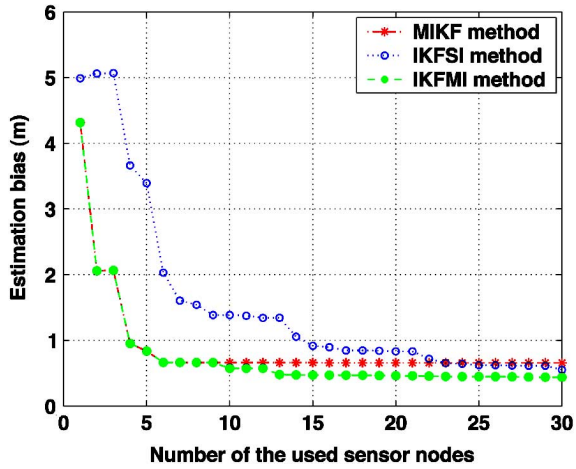


Fig. 6. Estimation bias versus number of selected sensor nodes for methods MIKF, IKFSI, and IKFMI.

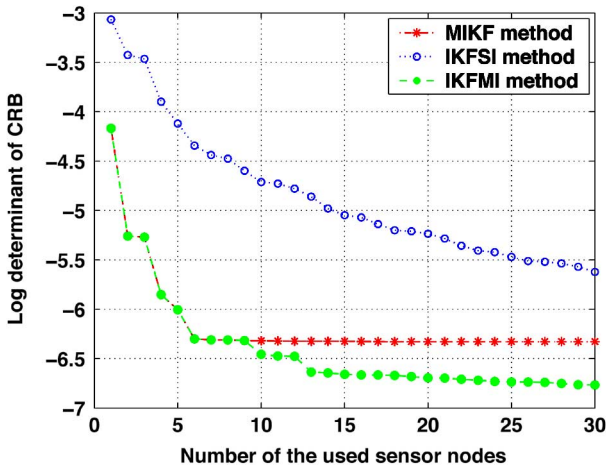


Fig. 7. Log determinant of the CRB versus number of selected sensor nodes for methods MIKF, IKFSI, and IKFMI.

IKFSI, even though in the MIKF we use only a single iteration over the selected sensor nodes. And the performance of the MIKF is worse than the IKFMI; however, the difference is not very large.

Note that the results from the above examples show not only the comparison between the performance but the comparison between the required communications to reach a performance threshold. According to our methods, the data transmission between neighbored sensor nodes is the same. Therefore, the number of data transmission hops approximates the required communications (we assume the sensor nodes are uniformly distributed in an area). For example, in Fig. 7, if we set the threshold of the log-determinant of the CRB to be -5.5, then the MIKF and IKFMI methods need four transmission hops to reach this performance threshold, but the IKFSI method needs 25 hops. This means the MIKF and IKFMI are more energy efficient than IKFSI.

To summarize the above results, when comparing the estimation performance of these four methods, we obtain the following order:

$$\text{IKFMI} > \text{MIKF} > \text{IKFSI} > \text{SIKF}.$$

That is, the performance of the iterative Kalman filtering is better than the extended Kalman filtering implementation, and the multiple local iteration is better than the single local iteration implementation. When comparing the required communications to reach their performance limits, we obtain

$$\text{IKFMI} = \text{IKFSI} \gg \text{MIKF} = \text{SIKF}.$$

Therefore, when considering both estimation performance and energy consumption, the MIKF method has the most advantages for applications in wireless sensor networks, since it can obtain a high processing performance using relatively low communication energy. By fully taking advantage of the local operation at each sensor node, we can obtain a substantial improvement in processing performance while still keeping energy consumption low.

VI. Conclusion

In this paper, we addressed the problem of developing an energy-efficient distributed estimation method for applications in wireless sensor networks. We first proposed a statistical measurement model in which we separated the linear and nonlinear parameters. This modelling strategy can reduce the corresponding processing complexity. Based on this model, we analyzed the implementation of the maximum likelihood estimation using the Gauss-Newton method and derived a basic framework of a fully distributed estimation method. This method is motivated by the analysis of incremental least squares methods by Bertsekas [25]. In our method, we decreased the energy usage while still maintaining an acceptable estimation performance. We also proposed three modifications of the basic distributed estimation method to improve processing performance. We introduced information-driven collaborative signal processing and proposed a sensor-node scheduling scheme to reduce further the communication energy consumption. Finally, we used numerical examples to study the performance of the proposed methods and determined that of the methods considered here the multiple iterations Kalman filtering method is the most suitable approach to be applied in the wireless sensor network, considering both the estimation accuracy and required communication burden. In future work, we will extend the proposed distributed estimation method to the dynamic signal model. We will study the distributed implementation of other numerical methods, such as the full Newton-type methods. We will also investigate the application of the proposed distributed method to solve additional practical problems.

Appendix
Proof of Theorem 4.1

Recall that the recursive equation (29) in the kth iteration of the Gauss-Newton method is obtained by linearizing the nonlinear signal model at the current estimate $\boldsymbol{\theta}^{(k)}$ and applying the linear least squares procedure to solve it. We denote the error function as $J = \| \tilde{\boldsymbol{y}} - \tilde{A}(\boldsymbol{\theta})\boldsymbol{x} \|^2 = \sum_{i=1}^{M} \| \tilde{\boldsymbol{y}}_i - \tilde{A}_i(\boldsymbol{\theta})\boldsymbol{x} \|^2$,

where $\|\cdot\|$ represents the Euclidean norm of a vector. Then we can linearize this error function at $\boldsymbol{\theta}^{(k)}$ as

$$\begin{aligned} J &= \sum_{i=1}^{M} \left\| \tilde{\boldsymbol{y}}_i - \tilde{A}_i(\boldsymbol{\theta})\boldsymbol{x} \right\|^2 \\ &\approx \sum_{i=1}^{M} \left\| \tilde{\boldsymbol{y}}_i - \tilde{A}_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} \right. \\ &\quad \left. - \left.\frac{\partial[\tilde{A}_i(\boldsymbol{\theta})\boldsymbol{x}]}{\partial \boldsymbol{\theta}^{\mathrm{T}}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\right) \right\|^2 \\ &= \sum_{i=1}^{M} \left\| \tilde{\boldsymbol{y}}_i - \tilde{A}_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} + D_i H_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta}^{(k)} \right. \\ &\quad \left. - D_i H_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta} \right\|^2 \\ &= \left\| \tilde{\boldsymbol{y}} - \tilde{A}\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} + DH\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta}^{(k)} - DH\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta} \right\|^2. \end{aligned} \tag{57}$$

Then, by applying the linear least squares procedure, we have

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{(k+1)} &= \arg\min_{\boldsymbol{\theta}} \left\| \tilde{\boldsymbol{y}} - \tilde{A}\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} \right. \\ &\quad \left. + DH\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta}^{(k)} - DH\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta} \right\|^2 \\ &= \left[H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1}H\left(\boldsymbol{\theta}^{(k)}\right)\right]^{-1} H^{\mathrm{T}}\left(\boldsymbol{\theta}^{(k)}\right)\Sigma^{-1} \\ &\quad \times \left(\boldsymbol{y} - A\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} + H\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta}^{(k)}\right) \end{aligned} \tag{58}$$

which is equivalent to (29).

Next, according to Proposition 1 proved in [25], we have the following results.

Assuming that matrix $C_1^{\mathrm{T}}C_1$ is positive definite, the linear least squares estimates

$$\boldsymbol{\psi}_i = \arg\min_{\boldsymbol{\theta}} \sum_{j=1}^{i} \|\boldsymbol{z}_j - C_j\boldsymbol{\theta}\|^2, \quad i = 1, \ldots, M \tag{59}$$

can be generated by the algorithm

$$\boldsymbol{\psi}_i = \boldsymbol{\psi}_{i-1} + \Gamma_i^{-1}C_i^{\mathrm{T}}(\boldsymbol{z}_i - C_i\boldsymbol{\psi}_{i-1}), \quad i = 1, \ldots, M \tag{60}$$

where $\boldsymbol{\psi}_0$ is an arbitrary vector, and the positive definite matrixes Γ_i are generated by

$$\Gamma_i = \Gamma_{i-1} + C_i^{\mathrm{T}}C_i, \qquad i = 1, \ldots, M \tag{61}$$

with $\Gamma_0 = 0$.

Comparing the linear least squares problem (59) with the linearized error function (57), we find that when we apply the identifications

$$\boldsymbol{z}_i = \tilde{\boldsymbol{y}}_i - \tilde{A}_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{x} + D_i H_i\left(\boldsymbol{\theta}^{(k)}\right)\boldsymbol{\theta}^{(k)} \tag{62}$$

$$C_i = D_i H_i\left(\boldsymbol{\theta}^{(k)}\right) \tag{63}$$

and let $\boldsymbol{\psi}_0 = \boldsymbol{\theta}^{(k)}$, we can generate $\boldsymbol{\theta}^{(k+1)}$ using the incremental algorithm in Theorem 4.1.

REFERENCES

[1] S. Kumar, F. Zhao, and D. Shepherd, Eds., "Special issue on collaborative signal and information processing in microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, Mar. 2002.

[2] I. F. Alyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[4] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA: Morgan Kaufmann, 2004.

[5] T. Zhao and A. Nehorai, "Localization of diffusive sources using distributed sequential Bayesian methods in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Toulouse, France, May 2006, vol. 4, pp. 985–988.

[6] L. Rossi, B. Krishnamachari, and C.C.J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. 1st IEEE Int. Conf. Sensor and Ad hoc Commun. Networks (SECON)*, Santa Clara, CA, Oct. 2004, pp. 460–469.

[7] Y. Levinbook and T. F. Wong, "Maximum likelihood diffusive source localization based on binary observations," in *Proc. 38th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2004, vol. 1, pp. 1008–1012.

[8] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug. 2003.

[9] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[10] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Particle filtering under communications constraints," in *Proc. IEEE/SP 13th Wkshp. Statist. Signal Process.*, Jul. 2005, pp. 89–94.

[11] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Optimization approaches to dynamic routing of measurements and models in a sensor network object tracking problem," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2005, vol. 5, pp. 1061–1064.

[12] M. Coates, "Distributed particle filters for sensor networks," in *3rd Int. Symp. Inf. Process. Sensor Netw. 26–27 Apr. 2004*, 2004, pp. 93–107.

[13] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Appl. Signal Process.*, vol. 4, pp. 378–391, Mar. 2003.

[14] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *Int. J. High-Performance Computing Applicat.*, vol. 16, no. 3, pp. 90–110, 2002.

[15] B. H. Cheng, R. E. Hudson, F. Lorenzelli, L. Vandenberghe, and K. Yao, "Distributed Gauss-Newton method for node localization in wireless sensor networks," in *Proc. IEEE 6th Wkshp. Signal Process. Adv. Wireless Commun.*, Jun. 2005, pp. 915–919.

[16] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2006, pp. 168–176.

[17] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2004, pp. 20–27.

[18] D. Blatt and A. Hero, "Distributed maximum likelihood estimation for sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2004, vol. 3, pp. 929–932.

[19] R. D. Nowak, "Distributed EM algorithm for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

[20] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.

[21] T. Zhao and A. Nehorai, "Detecting and estimating biochemical dispersion of a moving source in a semi-infinite medium," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2213–2225, Jun. 2006.

[22] A. Nehorai, B. Porat, and E. Paldi, "Detection and localization of vapor-emitting sources," *IEEE Trans. Signal Process.*, vol. SP-43, pp. 243–253, Jan. 1995.

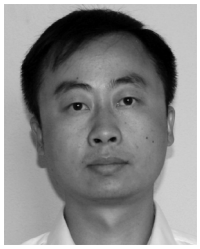
[23] A. R. Gallant, *Nonlinear Statistical Models*. New York: Wiley, 1987.

[24] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. New York: Wiley, 1989.

[25] D. P. Bertsekas, "Incremental least squares methods and the extended Kalman filter," *SIAM J. Optimiz.*, vol. 6, no. 3, pp. 807–822, 1996.

[26] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[27] R. P. Wishner, J. A. Tabaczynski, and M. Athans, "A comparison of three non-linear filters," *Automatica*, vol. 5, pp. 487–496, 1969.
[28] B. M. Bell and F. W. Cathey, "The iterated Kalman filter update as a Gauss-Newton Method," *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.
[29] B. M. Bell, "The iterated Kalman smoother as a Gauss-Newton method," *SIAM J. Optimiz.*, vol. 4, no. 3, pp. 626–636, Aug. 1994.
[30] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[31] X. Liao and L. Carin, "Application of the theory of optimal experiments to adaptive electromagnetic-induction sensing of buried targets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 8, pp. 961–972, Aug. 2004.



Tong Zhao (S'02–M'07) received the B.Eng. and M.Sc. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, in 1997 and 2000, respectively. He received the Ph.D. degree in electrical engineering from Washington University, Saint Louis, MO, in 2006.

Currently, he is a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, University of California, Davis. His research interests are adaptive and statistical signal processing, including detection and estimation theory, optimization, distributed signal processing, sequential Bayesian inference, Monte Carlo methods, etc., and their applications in sensor arrays, communications, wireless sensor networks, and biochemical sensors.

Arye Nehorai (S'80–M'83–SM'90–F'94) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion, Israel, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

From 1985 to 1995, he was a faculty member with the Department of Electrical Engineering, Yale University, New Haven, CT. In 1995, he joined the Department of Electrical Engineering and Computer Science, University of Illinois at Chicago (UIC), as Full Professor. From 2000 to 2001, he was Chair of the Department of Electrical and Computer Engineering (ECE) Division, which then became a new department. In 2001, he was named University Scholar of the University of Illinois. In 2006, he became Chairman of the Department of Electrical and Systems Engineering at Washington University, St. Louis, MO. He is the inaugural holder of the Eugene and Martha Lohman Professorship and the Director of the Center for Sensor Signal and Information Processing (CSSIP) at WUSTL since 2006.

Dr. Nehorai was Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2000 to 2002. From 2003 to 2005, he was Vice President (Publications) of the IEEE Signal Processing Society, Chair of the Publications Board, member of the Board of Governors, and member of the Executive Committee of this Society. From 2003 to 2006, he was the founding editor of the special columns on Leadership Reflections in the *IEEE Signal Processing Magazine*. He was corecipient of the IEEE SPS 1989 Senior Award for Best Paper with P. Stoica, coauthor of the 2003 Young Author Best Paper Award, and corecipient of the 2004 Magazine Paper Award with A. Dogandzic. He was elected Distinguished Lecturer of the IEEE SPS for the term 2004 to 2005 and received the 2006 IEEE SPS Technical Achievement Award. He is the Principal Investigator of the new multidisciplinary university research initiative (MURI) project entitled Adaptive Waveform Diversity for Full Spectral Dominance. He has been a Fellow of the Royal Statistical Society since 1996.