DISTRIBUTED TRACKING IN AD-HOC SENSOR NETWORKS

Tien Pham

US Army Research Laboratory
Acoustic Signal Processing Branch
Adelphi, MD 20783 USA

Haralabos C. Papadopoulos

Electrical and Computer Engineering Dept.
University of Maryland
College Park, MD 20742 USA

ABSTRACT

In this paper, we present distributed algorithms for tracking a moving source via an ad-hoc network of sensors. Tracking is performed by employing a Kalman filter at all detecting nodes in the network. The Kalman filter employed at any given node exploits the availability of source-location snapshot and prediction estimates, both of which are computed via distributed locally constructed algorithms over the ad-hoc network. As our brief simulation-based analysis reveals, the source-tracking performance of the proposed algorithms is a function of the motion dynamics of the source, the snapshot source-localization algorithm employed, the network topology, and the number of iterations employed in the distributed approximation algorithm.

1. INTRODUCTION

Interest in developing decentralized algorithms for data processing in ad-hoc sensor networks (AHSN) has grown considerably in recent years. For many data fusion problems that arise in such large-scale sensor networks, decentralized algorithms are preferable to their centralized counterparts, as they do not require a single central fusion center, or global knowledge of the sensor network topology. Furthermore, distributed data fusion algorithms are inherently modular, scalable, fault tolerant, and readily adaptable to changes in the sensor network topology [1].

In this paper, we consider the problem of decentralized tracking of a moving acoustic source in an AHSN. We assume that the nodes in the network collect source-node range measurements at a fixed sensing rate. As shown in Fig. 1, source tracking is achieved via a sequence of distributed computations performed over a sequence of detecting subnetworks in the AHSN, whereby the detecting subnetwork at any time consists of all nodes (in proximity to the source) with high enough signal-to-noise ratio (SNR) in their measurements. We develop distributed Kalman filter-based tracking algorithms, according to which, each step of the algorithm is implemented over every node in the detecting subnetwork. The viability of the proposed algorithms for performing distributed source tracking is suggested by a brief simulation-based performance evaluation.

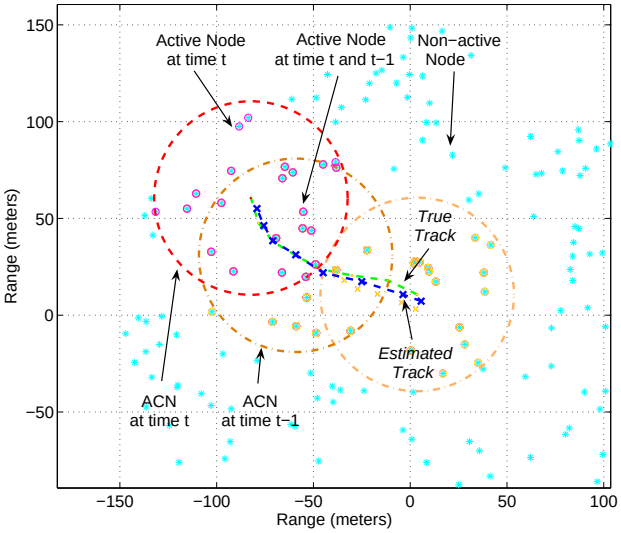

Fig. 1. Source tracking in a large-scale sensor network via computations over a sequence of subnetworks formed by the detecting nodes in the vicinity of the source.

2. SYSTEM MODEL

The setting of interest involves tracking the location of a single source as it moves across a large-scale sensor network, based on a sequence of snapshot measurements collected by sensor nodes across the network. We assume that, given a set of measurements at time t, a snapshot estimate of the source location is obtained by a source localization algorithm based on the data of the detecting nodes, *i.e.*, based on the data obtained by the subset of the nodes in the network in proximity to the source with high enough SNR data.

We first present a state-space model describing the motion dynamics of the source, and an observation model that is based on the snapshot estimates provided by the detecting nodes. We then present models for the topology of the overall network and the computation subnetworks employed in tracking the source location dynamics.

2.1. State-Space Model

We assume that the moving source has independent motion components in two dimensions and that the source motion in each dimension follows a constant velocity model with a random acceleration [2]. The model that we employ for developing tracking algorithms exploits the fact that, at any given time t, sensors in the sensor field obtain acoustic measurements, based on which they can form a source localization estimate of the acoustic source. In particular, letting $P_n(t)$, and $V_n(t)$ denote the position and velocity, respectively, of the acoustic source in the nth dimension ($n = 1,\ 2$) at time t, the tracking algorithms we develop view the resulting source position estimates, $Z_n(t)$, (obtained via a source localization algorithm at time t), as observations in a *single* measurement equation. We remark that the source-location estimate error sequences, $e_n(t) = Z_n(t) - P_n(t)$ (for $n = 1,\ 2$), are, in general, correlated in time (t). To account for this temporal correlation, we model $e_n(t)$ (for $n = 1,\ 2$) as a pth order autoregressive (AR(p)) process. It is assumed that the AR parameters of the process $e_n(t)$, *i.e.*, its order p, the $p \times 1$ vector $\mathbf{a}_p = \begin{bmatrix} a_p(1) & a_p(2) & \cdots & a_p(p) \end{bmatrix}^T$, and the associated innovation process power, σ_u^2, are first estimated during a training mode.

Letting $E_n(t) = \begin{bmatrix} e_n(t) & e_n(t-1) & \cdots & e_n(t-(p-1)) \end{bmatrix}^T$, the dynamics of the state $X_n(t) = \begin{bmatrix} P_n(t) & V_n(t) & E_n^T(t) \end{bmatrix}^T$ for $n = 1,\ 2$, are described by the following equation

$$X_n(t+1) = \mathbf{F}\, X_n(t) + \mathbf{G}\, B_n(t),\ t = 0, 1, \ldots \qquad \text{(1a)}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & T_s & \mathbf{0}_p^T \\ 0 & \rho & \mathbf{0}_p^T \\ \mathbf{0}_p & \mathbf{0}_p & \mathbf{A}_p \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} 0 & 0 \\ T_s & 0 \\ \mathbf{0}_p & \mathbf{u}_p \end{bmatrix}, \qquad \text{(1b)}$$

T_s denotes the snapshot update interval, $\mathbf{0}_k$ is $k \times 1$-vector of 0's, $\mathbf{u}_p = [1\ \mathbf{0}_{p-1}^T]^T$,

$$\mathbf{A}_p = \begin{bmatrix} & \mathbf{a}_p^T & \\ I_{(p-1)} & & \mathbf{0}_{p-1} \end{bmatrix}, \qquad \text{(1c)}$$

I_k is the $k \times k$ identity matrix, $B_n(t) = \begin{bmatrix} A_n(t) & U_n(t) \end{bmatrix}^T$, and where the acceleration process $A_n(t)$ and the snapshot-estimate error innovation process $U_n(t)$ are uncorrelated zero-mean white sequences, *i.e.*, $E\left[B_n(t) B_n^T(\tau)\right] = Q\delta[t-\tau]$, and where $Q = \mathrm{diag}(\sigma_a^2, \sigma_u^2)$. Finally, the parameter ρ in (1b) is given by $\rho = \sqrt{(\sigma_v^2 - \sigma_a^2\, T_s^2)/\sigma_v^2}$, where σ_v^2 denotes the variance of $V_n(t)$.

For each $n \in \{1, 2\}$, the associated (scalar) measurement equations are given by

$$Z_n(t) = \mathbf{H}\, X_n(t) = P_n(t) + e_n(t) \qquad \text{(2)}$$

where $\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & \mathbf{0}_{p-1}^T \end{bmatrix}$. We remark that the case where the snapshot error sequences $e_n(t)$ can be accurately modeled as white (*i.e.*, $p = 0$) is also captured by the model (1)–(2) by setting $p = 1$, $a_1(1) = 0$, and $\sigma_u^2 = E\left[e_n^2(t)\right]$.

2.2. Network Model

In this work we consider large-scale networks of uniformly distributed sensors. We focus on bidirectional network topologies, according to which, each node is assumed to establish bidirectional noise-free communication with a subset of nodes in its proximity. Letting N denote the total number of nodes in the network, the network topology is described by an $N \times N$ matrix Φ, where $\phi_{ij} = [\Phi]_{ij}$ denotes the connection status of the link between nodes $i \neq j$, defined as

$$\phi_{ij} = \phi_{ji} = \begin{cases} 1 & \text{if } i \leftrightarrow j \\ 0 & \text{otherwise} \end{cases} \qquad \text{(3a)}$$

and where $i \leftrightarrow j$ denotes that nodes i and j are bidirectionally connected. We also let for convenience [3]

$$\phi_{ii} = -\sum_{j \neq i}^{N} \phi_{ij}\,. \qquad \text{(3b)}$$

The connection status ϕ_{ij} of any two nodes i and j is modeled as a probabilistic function of d_{ij}, the distance between nodes i and j, and is given by

$$\Pr[\phi_{ij} = 1] = 2^{-\left(\frac{d_{ij}}{d_o}\right)^m}, \qquad \text{(4)}$$

where d_o denotes the nominal distance at which nodes i and j are connected with probability $\frac{1}{2}$, and where the parameter m determines the rate of decay of probability of connection with distance and typically satisfies $2 \leq m \leq 4$, [3].

The source tracking algorithms we consider are implemented sequentially over suitably chosen sequences of subnetworks. To this end, we let $\mathcal{I}(t)$ denote the set of nodes that comprise the active *computation* network (ACN) employed at time t, *i.e.*, the subnetwork over which the tth tracking estimate is to be computed. In this paper we focus on the simplest case where the set of nodes in the computation network at time t coincides with the subset of the nodes that detect the source at time t (*e.g.*, in Fig. 1). In general, however, the computation network may also include additional peripheral nodes to assist in the computation.

The network topology of the ACN at time t can be conveniently expressed in terms of an $N \times N$ matrix $\tilde{\Phi}(t)$ where the $(i,\ j)$th element of $\tilde{\Phi}(t)$, for $i \neq j$, is given by

$$\tilde{\phi}_{ij}(t) = \tilde{\phi}_{ji}(t) = \begin{cases} \phi_{ij} & \text{if } i, j \in \mathcal{I}(t) \\ 0 & \text{otherwise} \end{cases} \qquad \text{(5)}$$

while, for convenience, we set $\tilde{\phi}_{ii}(t) = -\sum_{j \neq i} \tilde{\phi}_{ij}(t)$.

Letting $M(t) = |\mathcal{I}(t)|$, and $\{\mathcal{I}(t)\}_i$ denote the ith element in $\mathcal{I}(t)$, the ACN network topology at time t can be alternatively described via $\mathcal{I}(t)$ (the subset of nodes in the computation network), and an $M(t) \times M(t)$ matrix $\Psi(t)$ where the $(i,\ j)$th element of $\Psi(t)$, for $i \neq j$, is defined as

$$\psi_{ij}(t) = \psi_{ji}(t) = \phi_{\{\mathcal{I}(t)\}_i\, \{\mathcal{I}(t)\}_j} \qquad \text{(6)}$$

while again, for convenience, we set $\psi_{ii}(t) = -\sum_{j\neq i}\psi_{ij}(t)$. We can think of the network described by $\tilde{\Phi}(t)$ in (5) as a large-scale network where the only available connections for performing computations are among the set of nodes in $\mathcal{I}(t)$. Alternatively, we can focus on the ACN formed by the nodes in $\mathcal{I}(t)$, with network topology given by $\Psi(t)$ in (6).

3. KALMAN FILTER TRACKING MODEL

We first consider the problem of tracking the location of the source with motion dynamics described by (1) at a fictitious node that has available at time t all the snapshot estimates of the source location up to time t, *i.e.*, $\{Z_n(\tau)\}_{\tau\leq t}$ given by (2). We assume that the parameters of the AR(p) process $e_n(t)$ used in the model (1) have been estimated via training. In particular, a sequence of autocorrelation sequence estimates $\hat{r}_e(i)$ for $0 \leq i \leq p$ is first obtained (based on a sufficiently large set of training sample paths) and, subsequently, estimates of $\mathbf{a}_p$ and σ_u^2 are obtained by exploiting the normal equations and the energy matching property, respectively, for AR modeling. Given that we also assume independent motion in each dimension, the 2-D source localization problem decouples into two independent 1-D source localization problems [4]. For any t and s, we let $\hat{X}_n(t|s)$ denote the linear mean-square-error (LMSE) estimate of the state $X_n(t)$ at time t based on all observations up to time step s (*i.e.*, $\{Z_n(\tau)\}_{\tau\leq s}$) and $\hat{\mathbf{\Sigma}}_n(t|s)$ denote the covariance matrix of the associated estimate. Evidently, the $(1,1)$ entry of $\hat{\mathbf{\Sigma}}_n(t|s)$ provides the mean-square-error (MSE) of the associated position estimate.

The Kalman filter (KF) provides a recursive algorithm for obtaining the LMSE estimate $\hat{X}_n(t|t)$ of the state $X_n(t)$ based on all snapshot estimates up to time t, in terms of $\hat{X}_n(t-1|t-1)$ and the new observation $Z_n(t)$. The KF algorithm takes the following form [4]:

$$\hat{X}_n(t|t-1) = \mathbf{F}\,\hat{X}_n(t-1|\,t-1) \tag{7a}$$

$$\mathbf{\Sigma}_n(t|t-1) = \mathbf{F}\,\mathbf{\Sigma}_n(t-1|\,t-1)\,\mathbf{F}^T + \mathbf{G}\,\mathbf{Q}\,\mathbf{G}^T \tag{7b}$$

$$\hat{X}_n(t|t) = \hat{X}_n(t|t-1) + \mathbf{K}_n(t)[Z_n(t) - \mathbf{H}\hat{X}_n(t|t-1)] \tag{7c}$$

$$\mathbf{\Sigma}_n(t|t) = \mathbf{\Sigma}_n(t|t-1) - \mathbf{K}_n(t)\,\mathbf{H}\,\mathbf{\Sigma}_n(t|t-1) \tag{7d}$$

$$\mathbf{K}_n(t) = \mathbf{\Sigma}_n(t|t-1)\mathbf{H}^T[\mathbf{H}\mathbf{\Sigma}_n(t|t-1)\mathbf{H}^T]^{-1}\,. \tag{7e}$$

The algorithm is initialized with

$$\hat{X}_n(0|0) = \begin{bmatrix} Z_n(0) \\ 0 \\ \mathbf{0}_p \end{bmatrix}, \mathbf{\Sigma}_n(0|0) = \begin{bmatrix} \hat{r}(0) & 0 & -\hat{\mathbf{r}}_p^T \\ 0 & \sigma_v^2 & \mathbf{0}_p^T \\ -\hat{\mathbf{r}}_p & \mathbf{0}_p & \hat{\mathbf{R}}_p \end{bmatrix},$$

where $\hat{\mathbf{r}}_p = [\hat{r}_e(0)\ \hat{r}_e(1)\ \ldots\ \hat{r}_e(p-1)]^T$, and $\hat{\mathbf{R}}_p$ is the $p \times p$ Toeplitz matrix of $\hat{\mathbf{r}}_p$.

4. DISTRIBUTED TRACKING

The KF algorithm (7) serves as a basis for developing distributed tracking algorithms, according to which the tth step of (7) is performed at *each* node within the ACN, $\{\mathcal{I}(t), \Psi(t)\}$. Observation of (7) reveals that for any node in $\mathcal{I}(t)$ to be able to perform the tth step of the algorithm and obtain $\hat{X}_n(t|t)$, the node must have available the gain $\mathbf{K}_n(t)$, the snapshot estimate $Z_n(t)$, and the previous tracking estimate $\hat{X}_n(t-1|t-1)$. As the gain sequence $\mathbf{K}_n(t)$ can be precomputed at each node, propagation of the index t as the source moves through the network suffices for allowing any node in $\mathcal{I}(t)$ to compute $\mathbf{K}_n(t)$. We next focus on how $Z_n(t)$ and $\hat{X}_n(t-1|t-1)$ can be approximated via distributed computations over the computation network $\{\mathcal{I}(t),\ \Psi(t)\}$.

4.1. Distributed Snapshot Estimate Computation

We next present a class of algorithms that are locally constructed over the ACN at time t and can be used to provide at each node in $\mathcal{I}(t)$ an approximation to a "target" source-localization estimate $Z_n(t)$. These algorithms are extensions of the source localization techniques in [5] that exploit improved versions of the distributed computation algorithms presented in [3]. We first present a (target) weighted centralized linear least-squares (LLS) type localization algorithm and, subsequently, develop distributed algorithms for approximating this algorithm at each node in the ACN.

4.1.1. Distributed Linear Least-Square Estimation

To illustrate the principles behind the distributed tracking algorithms we propose, we consider a weighted centralized LLS-type localization algorithm that is a generalized version of [5]. It is assumed that the ith node in the network knows its location, *i.e.*, it knows $\mathbf{p}_i = \begin{bmatrix} p_{i1} & p_{i2} \end{bmatrix}^T$, where p_{in} denotes the nth coordinate of the ith node location. It is also assumed that at time t the ith node possesses source-node range measurements of the form

$$S_i(t) = \sigma_{\mathrm{RSS}_i}(t) + \beta_i(t)$$

where the $\beta_i(t)$'s are zero-mean σ_β^2-power independent IID Gaussian sequences, and $\sigma^2_{\mathrm{RSS}_i}(t) = \sigma_s^2/r_i^2(t)$, with σ_s^2 and $r_i(t)$ denoting the (unknown) source power (received power at nominal distance 1) and the distance between the ith node and the source at time t, respectively. We remark that $r_i = \|\mathbf{p}_i - \mathbf{p}_{\mathrm{s}}\|$, where $\mathbf{p}_{\mathrm{s}}(t) = \begin{bmatrix} P_1(t) & P_2(t) \end{bmatrix}^T$, with $P_n(t)$ denoting the nth coordinate of the source location at time t. The source-location estimators we consider exploit the locally available minimum-variance unbiased estimates (MVUEs) of $\sigma^2_{\mathrm{RSS}_i}(t)$, *viz.*,

$$\widehat{\sigma^2}_{\mathrm{RSS}_i}(t) = S_i^2(t) - \sigma_\beta^2\,.$$

First, the ACN at time t is formed via threshold detection by including in the ACN only nodes with $\widehat{\sigma^2}_{\mathrm{RSS}_i}(t) > \sigma_T^2$, for some suitably preset threshold $\sigma_T^2 > 0$. For convenience, we omit the dependence of the estimator (and measurements) on t, as the source-location estimator at time t depends only on the measurements collected at time t.

Assuming M detecting nodes at time t, the LLS estimator of interest is based on $(M-1)$ range-squared difference equations, formed by viewing the M^{th} sensor (arbitrarily chosen) as a reference. In particular, the estimator exploits relative node-source range-squared estimates of the form

$$\widehat{\sigma^{-2}}_{\text{RSS}_i} = \left[\widehat{\sigma^2}_{\text{RSS}_i}\right]^{-1} = \frac{r_i^2}{\sigma_s^2} + \epsilon_i \tag{8}$$

and where the estimation-errors ϵ_i are independent in i. In simulation-based evaluation of the associated ith MSE, $\sigma_{\epsilon_i}^2 \triangleq E\left[\epsilon_i^2\right]$, reveals it is proportional to r_i^6, *i.e.*,

$$\sigma_{\epsilon_i}^2 \propto \sigma_{\text{RSS}_i}^{-6}\,, \tag{9}$$

for any threshold $\sigma_T^2 > 0$. By subtracting $\widehat{\sigma^{-2}}_{\text{RSS}_M}$ from $\widehat{\sigma^{-2}}_{\text{RSS}_i}$ for each $1 \le i \le M-1$, the (centralized) weighted LLS estimate of the set of unknowns, $\mathbf{x} \triangleq \begin{bmatrix} P_1 & P_2 & \sigma_s^2 \end{bmatrix}^T$, is then given by

$$\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{\Delta V}^{\text{T}}\mathbf{W}^{-1}\mathbf{\Delta V})^{-1}\mathbf{\Delta V}^{\text{T}}\mathbf{W}^{-1}\tilde{\mathbf{r}}\,, \tag{10}$$

where $\mathbf{\Delta V}$ is the following $(M-1)\times 3$ matrix

$$\mathbf{\Delta V} = \begin{bmatrix} (\mathbf{p}_1 - \mathbf{p}_M)^T & \widehat{\sigma^{-2}}_{\text{RSS}_1} - \widehat{\sigma^{-2}}_{\text{RSS}_M} \\ \vdots & \vdots \\ (\mathbf{p}_{M-1} - \mathbf{p}_M)^T & \widehat{\sigma^{-2}}_{\text{RSS}_{M-1}} - \widehat{\sigma^{-2}}_{\text{RSS}_M} \end{bmatrix} \tag{11}$$

$\mathbf{W}$ is the following $(M-1)\times(M-1)$ MSE-weight matrix

$$\mathbf{W} = \text{diag}(\sigma_{\epsilon_1}^2, \sigma_{\epsilon_2}^2, \cdots, \sigma_{\epsilon_{M-1}}^2) + \sigma_{\epsilon_M}^2\, \mathbf{1}\mathbf{1}^T\,, \tag{12}$$

and $\tilde{\mathbf{r}}_M$ is the following $(M-1)\times 1$ vector

$$\tilde{\mathbf{r}} = \frac{1}{2}\left[\|\mathbf{p}_1\|^2 - \|\mathbf{p}_M\|^2 \;\cdots\; \|\mathbf{p}_{M-1}\|^2 - \|\mathbf{p}_M\|^2\right]^T. \tag{13}$$

Due to (12) and (9), $\hat{\mathbf{x}}_{\text{LS}}$ from (10) is not a valid estimate as it depends on the unknown source location and power. A valid LLS-type estimate of the form (10) that is amenable to distributed implementation can be obtained by employing in (10) the following expression for $\mathbf{W}^{-1}$ in place of (12)

$$\hat{\mathbf{W}}^{-1} = \text{diag}(\widehat{\sigma^6}_{\text{RSS}_1}, \widehat{\sigma^6}_{\text{RSS}_2}, \cdots, \widehat{\sigma^6}_{\text{RSS}_{M-1}}) \tag{14}$$

where $\widehat{\sigma^6}_{\text{RSS}_i}$ is the MVUE of $\sigma_{\text{RSS}_i}^6$

$$\widehat{\sigma^6}_{\text{RSS}_i} = S_i^6 - 15\,\sigma_\beta^2\, S_i^4 + 45\,\sigma_\beta^4\, S_i^2 - 15\,\sigma_\beta^6\,. \tag{15}$$

Finally, the snapshot estimates $Z_1(t)$ and $Z_2(t)$ are given by the first two entries of the estimate vector $\hat{\mathbf{x}}_{\text{LS}}$ in (10).

4.1.2. Snapshot Estimate Distributed Implementation

We next describe the distributed algorithm employed to obtain an approximation to $Z_n(t)$ at each node in $\mathcal{I}(t)$. Distributed computation of $Z_n(t)$ relies on decomposing (10) with $\mathbf{W}^{-1}$ given by (14) into a set of parallel computations involving averages of local functions of the node data.

We first present the distributed algorithm for computing any such elementary averaging target computation. To this end, let the target (vector) computation be given by

$$\mathbf{g}(\mathbf{z}(t)) = \frac{1}{M(t)}\sum_{i=1}^{M(t)} \mathbf{f}_i(\mathbf{z}_i(t)) \tag{16}$$

where $M(t) = |\mathcal{I}(t)|$, $\mathbf{z}(t) = [\mathbf{z}_1^T(t)\; \mathbf{z}_2^T(t) \ldots\; \mathbf{z}_{M(t)}^T(t)]^T$, $\mathbf{z}_i(t)$ denotes the vector data at the ith node in $\mathcal{I}(t)$, and the $\mathbf{f}_i(\cdot)$'s are arbitrary local vector-valued functions. Dropping for convenience the parameter dependence on t, we consider a class of iterative algorithms that generate a sequence of approximations $\hat{\mathbf{b}}_i[k]$ to $\mathbf{g}(\mathbf{z})$ at the ith node via

$$\hat{\mathbf{b}}_i[k] = \begin{cases} \mathbf{f}_i(\mathbf{z}_i) & \text{if } k=0 \\ \hat{\mathbf{b}}_i[0] + \sum_j \rho_{ij}\hat{\mathbf{b}}_j[0] & \text{if } k=1 \\ (1+c)\left\{\hat{\mathbf{b}}_i[1] + \sum_j \rho_{ij}\hat{\mathbf{b}}_j[1]\right\} & \\ \quad -c\left\{\hat{\mathbf{b}}_i[0] + \sum_j \rho_{ij}\hat{\mathbf{b}}_j[0]\right\} & \text{if } k=2 \\ (1+c)\left\{\hat{\mathbf{b}}_i[k-1] + \sum_j \rho_{ij}\hat{\mathbf{b}}_j[k-1]\right\} & \\ \quad -c\hat{\mathbf{b}}_i[k-2] & \text{if } k>2 \end{cases} \tag{17}$$

where $0 \le c < 1$ is to be macroscopically selected, and where the ρ_{ij}'s are to be selected via local negotiations over Ψ. For a rule of the form (17) to be implementable over a topology Ψ, we must have $\rho_{ij} = \rho_{ji} = 0$ for any i and j with $\psi_{ij} = 0$. Assuming Ψ is a connected topology, proper choice of the ρ_{ij}'s in (17) yields converging approximations to (16). In this paper we exploit the following set of conditions that guarantee convergence of $\hat{\mathbf{b}}_i[k]$ to $\mathbf{g}(\mathbf{z})$:

$$\rho_{ij} = \rho_{ji} > 0\,, \quad \text{if } \psi_{ij} = 1 \tag{18a}$$

$$\rho_{ii} = -\sum_{j\neq i}\rho_{ij} \tag{18b}$$

$$|\rho_{ii}| < 1 \tag{18c}$$

$$0 \le c < 1 \tag{18d}$$

These conditions are a generalization of those presented in [3] and their sufficiency can be proved by exploiting Gersgorin's Theorem ([6], pp. 344–348). Although all $c \in [0,\, 1)$ yield convergence, as shown in [3], proper choice of c can greatly expedite the convergence rate of the computation.

Choices for the ρ_{ij}'s that satisfy (18) can be made via local negotiations. In particular, the following iterative local negotiation (LN) algorithm yields ρ_{ij} sets that outperform those reported in [3]. The algorithm yields sequences of improving sets of ρ_{ij}'s each of which satisfies (18). The algorithm achieves (18c) by guaranteeing that ρ_{ii} is at most $1-\epsilon$ for some small $\epsilon > 0$. At the outset, the algorithm is initialized with $\rho_{ij}^{(0)} = 0$ and $\psi_{ii}^{(0)} = |\psi_{ii}|$. Given an

arbitrarily small $\epsilon > 0$, the kth step of the algorithm at the ith node, for any $k \geq 1$, takes the following form:

(i) set and broadcast:

$$\delta_i^{(k)} = \begin{cases} (1-\epsilon+\rho_{ii}^{(k-1)})/\psi_{ii}^{(k)} & \text{if } \psi_{ii}^{(k)} > 0 \text{ and} \\ & 1-\epsilon+\rho_{ii}^{(k-1)} > 0 \\ 0 & \text{otherwise;} \end{cases}$$

(ii) for all j, set

$$\rho_{ij}^{(k)} = \begin{cases} \rho_{ij}^{(k-1)} + \min\{\delta_i^{(k)}, \delta_j^{(k)}\} & \text{if } \psi_{ij} = 1 \\ 0 & \text{otherwise;} \end{cases}$$

(iii) set

$$\rho_{ii}^{(k)} = -\sum_{j \neq i} \rho_{ij}^{(k)},$$
$$\psi_{ii}^{(k)} = |\{j;\, \psi_{ij} = 1,\, \delta_j^{(k)} > 0\}|\,.$$

It is straightforward to verify that this algorithm satisfies the conditions in (18) at every step k, and terminates after a finite number of iterations for any finite-size network.

The distributed algorithm (17) can be used to obtain an approximation to $Z_n(t)$ at every node in the ACN, $\mathcal{I}(t)$. In particular, using (14) and exploiting

$$\frac{\mathbf{\Delta V}^T \hat{\mathbf{W}}^{-1} \mathbf{\Delta V}}{M-1} = \frac{1}{M-1} \sum_{j=1}^{M-1} \widehat{\sigma^6}_{\mathrm{RSS}_j} (\mathbf{v}_j - \mathbf{v}_M)(\mathbf{v}_j - \mathbf{v}_M)^T \tag{19a}$$

$$\frac{\mathbf{\Delta V}^T \hat{\mathbf{W}}^{-1} \tilde{\mathbf{r}}_M}{M-1} = \frac{1}{M-1} \sum_{j=1}^{M-1} \widehat{\sigma^6}_{\mathrm{RSS}_j} \tilde{r}_j (\mathbf{v}_j - \mathbf{v}_M)\,, \tag{19b}$$

where $\mathbf{v}_i = \begin{bmatrix} \mathbf{p}_i^T & \widehat{\sigma^{-2}}_{\mathrm{RSS}_i} \end{bmatrix}^T$ and $\tilde{r}_j = \{\tilde{\mathbf{r}}\}_i$, and assuming the reference sensor vector $\mathbf{v}_M$ has been made available (via broadcasting) to all nodes in $\mathcal{I}(t)$, $\hat{\mathbf{x}}_{\mathrm{LS}}$ in (10) can be decoupled into parallel computations of weighted averages of the form (16). Distributed implementation of $\hat{\mathbf{x}}_{\mathrm{LS}}$ in (10) via (19) involves six parallel approximations for (19a) and three parallel approximations for (19b). In particular, the snapshot-estimate computation at the ith node in the ACN takes the following form: (i) apply the distributed algorithm (17) for k_o iterations (with k_o sufficiently large), locally initialized via $\hat{\mathbf{b}}_i[0] = \begin{bmatrix} \hat{b}_i^{(1)}[0] & \hat{b}_i^{(2)}[0] & \cdots & \hat{b}_i^{(9)}[0] \end{bmatrix}^T$, where

$$\begin{bmatrix} \hat{b}_i^{(1)}[0] & \hat{b}_i^{(2)}[0] & \hat{b}_i^{(3)}[0] \\ \hat{b}_i^{(2)}[0] & \hat{b}_i^{(4)}[0] & \hat{b}_i^{(5)}[0] \\ \hat{b}_i^{(3)}[0] & \hat{b}_i^{(5)}[0] & \hat{b}_i^{(6)}[0] \end{bmatrix} = \widehat{\sigma^6}_{\mathrm{RSS}_1} (\mathbf{v}_i - \mathbf{v}_M)(\mathbf{v}_i - \mathbf{v}_M)^T \tag{20a}$$

and

$$\begin{bmatrix} \hat{b}_i^{(7)}[0] & \hat{b}_i^{(8)}[0] & \hat{b}_i^{(9)}[0] \end{bmatrix}^T = \widehat{\sigma^6}_{\mathrm{RSS}_1} \tilde{r}_i (\mathbf{v}_i - \mathbf{v}_M)\,; \tag{20b}$$

(ii) use the resulting approximations $\{\hat{b}_i^{(j)}[k_o]\}$ at the ith node to locally approximate the LLS estimate $\hat{\mathbf{x}}_{\mathrm{LS}}$ from (10). We remark that, due to the exponentially fast convergence characteristics of (17) [3], a small number of iterations of the distributed algorithm (17) suffice for providing very accurate approximations to $Z_n(t)$ at each node in $\mathcal{I}(t)$.

4.2. Prediction Estimates

$\hat{X}_n(t-1|t-1)$ is directly available only to the nodes in $\mathcal{I}(t)$ that were also part of the active network at time $t-1$, *i.e.*, only to the nodes in $\mathcal{I}(t) \cap \mathcal{I}(t-1)$. Given that the "measurements" employed at time $t-1$ are approximations to $Z_n(t-1)$, each of the resulting $\hat{X}_n(t-1|t-1)$ via (7) at the nodes in $\mathcal{I}(t) \cap \mathcal{I}(t-1)$ are in general distinct approximations. For this reason, an algorithm of the form (17) is employed on the network topology induced by the restriction of Φ on $\mathcal{I}(t) \cap \mathcal{I}(t-1)$, to provide to all the nodes in $\mathcal{I}(t) \cap \mathcal{I}(t-1)$ an approximation to the average of the available $\hat{X}_n(t-1|t-1)$ estimates. In parallel, a broadcasting algorithm is employed to provide $\hat{X}_n(t-1|t-1)$ estimates to the remaining nodes in $\mathcal{I}(t)$, according to which at each cycle, each node in the subset broadcasts its $\hat{X}_n(t-1|t-1)$ estimate (if one is available) to its neighbors, and iteratively computes a new estimate as the average of the available estimates.

5. SIMULATIONS

In this section we present a brief simulation-based performance evaluation of the proposed distributed tracking algorithms. In our simulations we employ as our figure of merit the sample-mean MSE performance of the associated localization and tracking algorithms based on 1000 independent realizations. In each realization, a network is first generated with $N = 400$ nodes uniformly distributed in a circle of radius $R_o = 200$ m. A network topology is then generated according to (4) with $m = 2$ and $d_o = 55$ m. At the outset (*i.e.*, prior to tracking) a set of base ρ_{ij}'s are computed for the entire network by applying the LN algorithm of Sec. 4.1.2 for 20 iterations. Next, a source-motion sample path is generated by placing the source at the origin of the network and using model (1) to generate a trajectory. Throughout the simulations we use SNR $= \sigma_s^2/\sigma_\beta^2 = 55.6$ dB, and employ a detection threshold, σ_T^2, of 20 dB. This threshold yields ACNs with (detecting) nodes within approximately a 60 m radius from the source location. Prior to running the distributed algorithm (17) on any given set of snapshot data, the base ρ_{ij}'s (computed at the outset on the whole network) are refined for 5 iterations by applying the LN algorithm on the current ACN. The distributed algorithm (17) is then applied with $c = 0.6$ for $k_o = 20$ iterations to approximate $Z_n(t)$ in (2) at each node in the ACN. Finally, $e(t)$ in (1) is modeled throughout as an AR(3) pro-

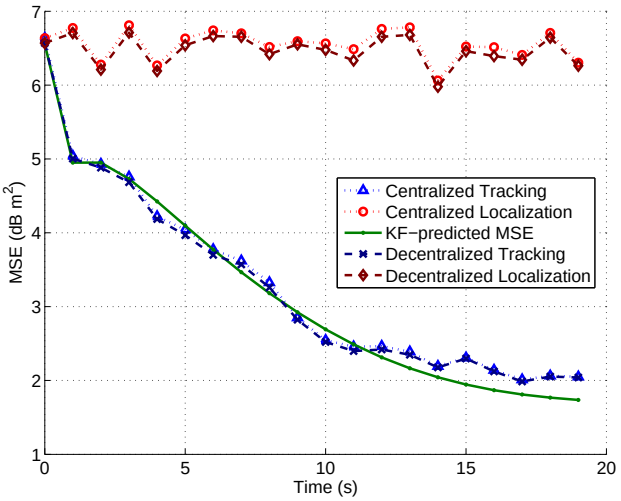

Fig. 2. MSE comparison of centralized and decentralized localization and tracking algorithms for $\sigma_v = 2$ m/s, $\sigma_a = 0.1$ m/s^2, and $T_s = 1$ s.

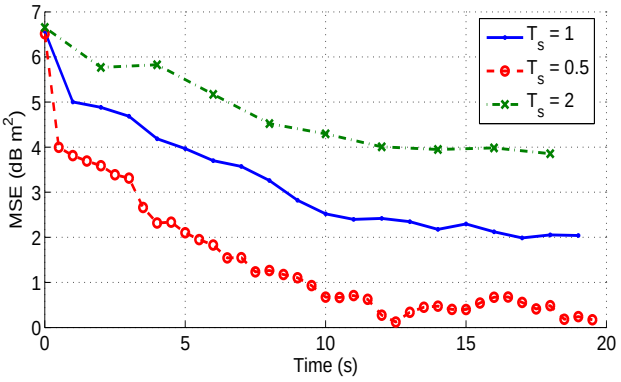

Fig. 3. MSE performance of distributed tracking algorithms for various values of T_s, for $\sigma_v = 2$ m/s and $\sigma_a = 0.1$ m/s^2.

cess. Fig. 2 depicts the simulated MSE performance of centralized and decentralized tracking algorithms in the case that $\sigma_v = 2$ m/s, $\sigma_a = 0.1$ m/s^2, and $T_s = 1$ s. As the figure suggests, the distributed tracking algorithms provide effectively the same MSE performance as their centralized counterparts, using a small number of iterations for initialization and distributed computation. Furthermore, distributed tracking yields a gain of approx. 4 dB with respect to the associated distributed snapshot estimation algorithm. Finally, we note that the tracking algorithm MSE performance is in close agreement with the MSE performance predicted by the $(1,\ 1)$ entry of the LHS of (7d).

Fig. 3 shows the MSE performance of the proposed algorithms as a function of the snapshot rate, when $\sigma_v = 2$ m/s and $\sigma_a = 0.1$ m/s^2. In particular, the successively lower curves show the simulated MSE of the distributed tracking

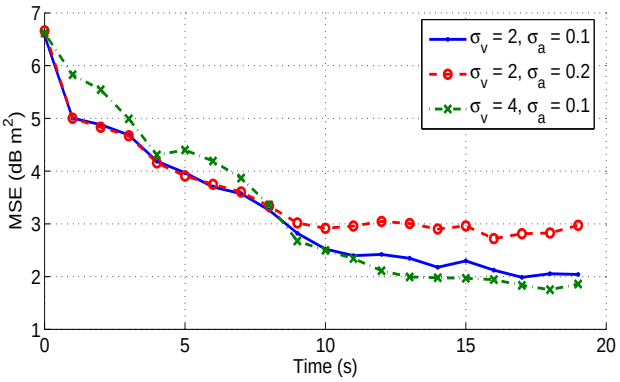

Fig. 4. MSE performance of distributed tracking algorithms as σ_v and σ_a are varied, while keeping $T_s = 1$ s.

algorithms for $T_s = 0.5, 1$, and 2 s, respectively. As the figure suggests, in this example, increasing the snapshot rate by a factor of 2 reduces the MSE by approx. 1.5 dB.

Fig. 4 shows the MSE performance of the distributed tracking algorithms as the source speed and acceleration parameters are varied, with snapshots taken at rate of 1 measurement/s per sensor. As the figure reveals, increasing σ_a by a factor of 2 while keeping σ_v unchanged results in increasing the steady-state MSE by 0.5 dB, while increasing σ_v by a factor of 2 while keeping σ_a unchanged does not appreciably affect the steady-state MSE performance.

6. REFERENCES

[1] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, Prentice Hall, 1st edition, 1995.

[2] Y. Bar-Shalom, X. Li, and T. Kirubargjan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc, New York, NY, 2001.

[3] D. S. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE J. Selected Areas Commun.*, vol. 23, no. 4, pp. 776–787, Apr. 2005.

[4] B. Anderson and J. Moore, *Optimal Filtering*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1st edition, 1979.

[5] T. Pham, D. S. Scherber, and H. C. Papadopoulos, "Distributed source localization algorithms for acoustic ad-hoc sensor networks," in *Proc. IEEE Work. Sensor Array Multichannel Signal Processing*, Spain, July 2004.

[6] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1st edition, 1979.