This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Cooperative Network Localization Via Node Velocity Estimation

Liang Dong
Department of Electrical and Computer Engineering
Western Michigan University, Kalamazoo, MI 49008 USA

***Abstract*—This paper addresses cooperative localization for mobile ad-hoc networks that benefits from the node velocity estimation. Given pair-wise range measurement and relative speed measurement between communicating nodes, the relative node positions are estimated using an extended Kalman filter. The state-space equation of the Kalman filter incorporates the node positions with their velocities. The measurement equation takes into account the log-normal distribution of the received signal power and the Gaussian distribution of the relative speed measurement error. Distributed algorithm is derived for practical use. The simulation results show the performance of the network localization with the assistance of node velocity estimation. The velocities are, however, not tracked using the Kalman filter; Separated method is proposed to estimate the node velocities.**

I. Introduction

In mobile ad-hoc networks, cooperative localization provides the relative coordinates of the wireless nodes based on the range measurements between nodes [1]. Range measurement usually relies on physical parameters such as received signal strength (RSS), angle of arrival (AOA), time of arrival (TOA), and time-distance of arrival (TDOA). The estimation inaccuracy is largely caused by the unreliable measurements due to node irregular coverage [2]–[4]. Multidimensional scaling (MDS) can be used for localization but the algorithm is prone to error in range measurement [5], [6]. Given the log-normal distribution of the RSS, or the Gaussian distribution of the TOA, a maximum likelihood estimator (MLE) can be used, trading complexity with accuracy [7]. However, a good initial guess is imperative for the MLE that applies fast gradient algorithms. A theoretical foundation for the problem of network localization was provided in [8]. It gives conditions for unique network localizability and the computational complexity of network localization.

Besides relative node positions, relative node velocities (speed and direction) provide important information in mobile ad-hoc networks. For example, the estimation of node location and velocity is crucial for a group of mobile robots for collision avoidance, where GPS is unavailable or too expensive to implement on each single unit. The mobile speed can be estimated using the maximum Doppler spread of the received signal in the multipath fading environment, or simply by a Doppler sensor if the dominant line-of-sight component is present. In this paper, the velocity estimation is applied to assist the cooperative network localization.

An extended Kalman filter was used in [9] to solve for GPS tracking in urban canyon environment. In [10], the distributed Kalman filter was reduced to two dynamic consensus problems on weighted measurements and inverse-covariance matrices. In [11], local Kalman filters were implemented on the reduced-dimensional subsystems in order to estimate the state of a sparsely connected large-scale system. The centralized Riccati and Lyapunov equations were computed iteratively with local communications, and the observations common among the local Kalman filters were fused. Along these lines, we develop an extended Kalman filtering method that incorporates node positions with their velocities in the state-space model. The estimation of node velocity aids the cooperative network localization, such that the localization performance improves over Kalman filter iterations. However, it can not track the node velocity variation. One needs higher order state vectors, or a separated algorithm to track the node velocity.

II. System Model

Consider a D-dimensional mobile ad-hoc network, e.g., $D = 2$ or 3. There are N mobile nodes in the network with unknown positions and velocities. There are M nodes with known positions and velocities, which are the M reference nodes. The position of the ith node is denoted as $\mathbf{x}_i$, and the velocity of the ith node is the time derivative $\dot{\mathbf{x}}_i$. $\mathbf{x}_i, \dot{\mathbf{x}}_i \in \mathbb{R}^D$. The cooperative localization and velocity estimation problem is to find the position and velocity vectors of the N unknown nodes, given the M reference nodes and the pair-wise scalar measurements. These pair-wise scalars include the range between a pair of nodes, and the relative speed of the nodes which is the relative velocity projected onto the line connecting that pair of nodes. The range between the ith and the jth nodes is given by the Euclidean distance as

$$d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)} \tag{1}$$

and the relative speed is given by

$$s_{ij} = \hat{\mathbf{x}}_{ij}^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j) \tag{2}$$

where $\hat{\mathbf{x}}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)/||\mathbf{x}_i - \mathbf{x}_j||$. The relative speed can be measured by the Doppler frequency shift in the received RF signal using a Doppler sensor. With the ith node being the receiver and the jth node the transmitter, the Doppler frequency shift is given by

$$f_d = \frac{\hat{\mathbf{x}}_{ij}^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)}{C} f_c \tag{3}$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

where, f_c is the universal carrier frequency in the network, C is the speed of electromagnetic wave propagation. With low mobile speed, the Doppler may not be measurable. Given consecutive node positions, we can obtain the relative speed of the nodes as the distance differential $\dot{d}_{ij}$, which is given by

$$\dot{d}_{ij} = \frac{(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||}(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j). \tag{4}$$

III. Cooperative Network Localization

A. Node Location Estimation

The node localization problem is the estimation of the coordinates $\{\mathbf{x}_i\}_{i=1}^N$ given the pair-wise range measurements. As a stand-alone problem, we assume that the node velocities are zeros, or we analyze a snapshot of the network such that the node positions do not change during the process time.

Let us use the classic MDS algorithm as an example. With the range measurement between the ith and the jth nodes, d_{ij}, the squared Euclidean distance is given by

$$d_{ij}^2 = \mathbf{x}_i^T\mathbf{x}_i - 2\mathbf{x}_i^T\mathbf{x}_j + \mathbf{x}_j^T\mathbf{x}_j. \tag{5}$$

The squared distance matrix, $\mathbf{D} = \{d_{ij}^2\}_{i,j=1}^N$, can be written as

$$\mathbf{D} = \psi\mathbf{e}^T - 2\mathbf{X}^T\mathbf{X} + \mathbf{e}\psi^T \tag{6}$$

where $\psi = [\mathbf{x}_1^T\mathbf{x}_1, \cdots, \mathbf{x}_N^T\mathbf{x}_N]^T$, $\mathbf{X}$ is the coordinate matrix given by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, and $\mathbf{e}$ is the N-dimensional vector of all ones. We have

$$\mathbf{A} = -\frac{1}{2}\mathbf{\Upsilon}\mathbf{D}\mathbf{\Upsilon} = \mathbf{\Upsilon}\mathbf{X}^T\mathbf{X}\mathbf{\Upsilon} \tag{7}$$

where $\mathbf{\Upsilon} = (\mathbf{I} - \mathbf{e}\mathbf{e}^T/N)$. Performing singular value decomposition (SVD) on matrix $\mathbf{A}$, we have

$$\mathbf{A} = \mathbf{U}\mathrm{diag}(\lambda_1, \ldots, \lambda_N)\mathbf{U}^T. \tag{8}$$

Since $||\mathbf{\Upsilon}||_2 = 1$, i.e., the largest singular value of $\mathbf{\Upsilon}$ is 1, the coordinate matrix can be calculated up to a translation and orthogonal transformation by

$$\mathbf{X} = \mathrm{diag}(\lambda_1^{1/2}, \ldots, \lambda_N^{1/2})\mathbf{U}^T. \tag{9}$$

The translation and the orthogonal transformation of $\mathbf{X}$ can be determined with sufficient reference nodes. The above derivation requires the knowledge of all the pair-wise distances as in matrix $\mathbf{D}$. This is unlikely in the network with power or bandwidth constraints. The distributed MDS was developed in [5] that depends only on local distances measurements. It reduced the complexity of the computation of matrix $\mathbf{A}$ and its SVD, with the assumption that the pair-wise distance graph is globally rigid and the relative map is uniquely determined.

B. Node Velocity Estimation

The node velocity estimation problem is to find the instant velocity of each node relative to a network group velocity, given the coordinates of the nodes and the pair-wise relative speeds $\{s_{ij}\}$. The relative speed can be obtained by measuring Doppler frequency shift or by differentiating the distance. This information of relative velocities (speeds and directions) can assist, for example, a fleet of unmanned aerial vehicles that fly in proximity to calibrate their inertial navigation systems, in order to avoid collision in GPS-denied environment.

Suppose there are N communicating nodes within the network. With $\mathbf{x}_i$'s known, $\hat{\mathbf{x}}_{ij}$'s can be calculated. If $\dot{\mathbf{x}}$ is a D-dimensional vector, there are DN unknowns. Suppose that every pair-wise Doppler is measured, a total of $\frac{N(N-1)}{2}$ s_{ij}'s are present with $1 \le i < j \le N$. A set of linear equations can be formulated as

$$\mathbf{R}\mathbf{v} = \mathbf{s} \tag{10}$$

where

$$\mathbf{R} = \left[\begin{array}{ccccccc}
\hat{\mathbf{x}}_{12}^T & -\hat{\mathbf{x}}_{12}^T & 0 & & & \cdots & 0 \\
\hat{\mathbf{x}}_{13}^T & 0 & -\hat{\mathbf{x}}_{13}^T & 0 & & \cdots & 0 \\
\hat{\mathbf{x}}_{14}^T & 0 & 0 & -\hat{\mathbf{x}}_{14}^T & 0 & \cdots & 0 \\
\vdots & & & & & & \vdots \\
\hat{\mathbf{x}}_{1N}^T & 0 & & \cdots & & 0 & -\hat{\mathbf{x}}_{1N}^T \\
0 & \hat{\mathbf{x}}_{23}^T & -\hat{\mathbf{x}}_{23}^T & 0 & & \cdots & 0 \\
0 & \hat{\mathbf{x}}_{24}^T & 0 & -\hat{\mathbf{x}}_{24}^T & 0 & \cdots & 0 \\
\vdots & & & & & & \vdots \\
0 & & \cdots & & 0 & \hat{\mathbf{x}}_{(N-1)N}^T & -\hat{\mathbf{x}}_{(N-1)N}^T
\end{array}\right],$$

$$\begin{aligned}
\mathbf{v} &= [\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \dot{\mathbf{x}}_3, \cdots, \dot{\mathbf{x}}_{N-1}, \dot{\mathbf{x}}_N]^T, \quad \text{and} \\
\mathbf{s} &= [s_{12}, s_{13}, s_{14}, \cdots, s_{1N}, s_{23}, s_{24}, \cdots, s_{(N-1)N}]^T.
\end{aligned}$$

The dimension of matrix $\mathbf{R}$ is $\frac{N(N-1)}{2} \times DN$ and its rank is given by

$$\mathcal{R}(\mathbf{R}) = \min\left(\frac{N(N-1)}{2}, DN - \frac{D(D+1)}{2}\right). \tag{11}$$

The rank of $\mathbf{R}$ reflects the $D(D+1)/2$ degrees of freedom of the velocity vectors that would not affect the Doppler measurement results.

In the two-dimensional case ($D = 2$), suppose that nodes 1 and 2 are reference nodes with $\dot{\mathbf{x}}_1$ and one component of $\dot{\mathbf{x}}_2$, e.g., $\dot{x}_{2x}$, known. The linear equations can be modified as in (12), where $\mathbf{R}'$ is a $(\frac{N(N-1)}{2} - 1) \times (2N - 3)$ matrix. (Please see at the bottom of the following page.) It is a tall matrix and guarantees full rank when $N \ge 4$. Therefore, the velocity vectors can be calculated as

$$\mathbf{v}' = \mathbf{R}'^{\dagger}\mathbf{s}' \tag{13}$$

where $\dagger$ denotes Pseudo-inverse. Since $\mathbf{R}'$ is a sparse matrix, the computation of the inverse of $\mathbf{R}'$ can be made efficient. When the velocities of two nodes, $\mathbf{v}_1$ and $\mathbf{v}_2$, are known, the algorithm for $D = 2$ can uniquely determine the velocities of any additional nodes as $N \ge 3$.

C. Network Localization Via Velocity Estimation

There are two issues that concern the network localization and velocity estimation in practice. First, the range measurement and the relative speed measurement are prone to error and impaired by the channel fading effect in the mobile communication scenario. When the MDS algorithm is used, no model assumptions are made about the statistical behavior of the observed dissimilarities (the received signal power or the time-of-arrival). Secondly, the measurements are distributed among the nodes in the network. A distributed algorithm is required to efficiently estimate the node location and velocity. The centralized estimation becomes impossible to implement as the scale of the network increases. The decentralized algorithm requires the range and relative speed measurements between a node and a small number of its neighbors, in stead of all $(M+N)(M+N-1)/2$ pair-wise measurements.

1) Centralized Extended Kalman Filter Approach: Let us start with a centralized algorithm that uses all pair-wise measurements. When each node is in low dynamic motion, the velocity can be modeled as a random-walk process, and the position is obtained by integrating the velocity. We consider the channel fading effect in network communication links, and use the extended Kalman filter (EKF) for the nonlinear model that has inputs of both range and relative speed measurements. The continuous-time state-space model is given by

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w} \tag{14}$$

where
$$\begin{aligned}
\mathbf{x} &= [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T, \dot{\mathbf{x}}_1^T, \dot{\mathbf{x}}_2^T, \ldots, \dot{\mathbf{x}}_N^T]^T, \\
\mathbf{F} &= \begin{bmatrix} \mathbf{0}_{DN\times DN} & \mathbf{I}_{DN\times DN} \\ \mathbf{0}_{DN\times DN} & \mathbf{0}_{DN\times DN} \end{bmatrix}, \\
\mathbf{w} &= [0, \ldots, 0, \mathbf{w}_{v1}^T, \ldots, \mathbf{w}_{vN}^T]^T.
\end{aligned}$$

$\mathbf{x}$ is the state vector of length $2DN$. $\mathbf{w}$ is uncorrelated zero-mean white-noise process with correlation matrix $\mathbf{Q}$ that models the random variations in the velocities. The state-space model is linear, whereas the measurement equations are nonlinear. The measurement equations are given by

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \tag{15}$$

where, $\mathbf{y}$ is a length $N(N-1)$ vector as $\mathbf{y} = [p_{12}, p_{13}, \ldots, p_{(N-1)N}, s_{12}, s_{13}, \ldots, s_{(N-1)N}]^T$. We adopt the shadowing channel model of log-normally distributed received signal power [12]. The received power measurements in dBm, $\{p_{ij}\}_{i<j}$, are given by

$$p_{ij} = P_0 - 10 n_p \log_{10} \frac{||\mathbf{x}_i - \mathbf{x}_j||}{d_0} + u_{ij} \tag{16}$$

where $u_{ij} \sim \mathcal{N}(0, \sigma_{dB}^2)$. The relative speed measurements, $\{s_{ij}\}_{i<j}$, are given by

$$s_{ij} = \frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)}{||\mathbf{x}_i - \mathbf{x}_j||} + v_{ij}. \tag{17}$$

We have $\mathbf{v} = [u_{12}, u_{13}, \ldots, u_{(N-1)N}, v_{12}, v_{13}, \ldots, v_{(N-1)N}]^T$. We assume that $\mathbf{v}$ is uncorrelated zero-mean white-noise process with correlation matrix $\mathbf{Q}_2$, and $\mathbf{v}$ is uncorrelated with $\mathbf{w}$. The corresponding discrete-time state-space model and measurement equations are given by

$$\begin{aligned}
\mathbf{x}(n+1) &= \mathbf{\Phi}\mathbf{x}(n) + \mathbf{w}_d(n) && (18)\\
\mathbf{y}(n) &= \mathbf{h}(\mathbf{x}(n)) + \mathbf{v}(n) && (19)
\end{aligned}$$

where $\mathbf{\Phi} = e^{\mathbf{F}T_s} = \mathbf{I} + \mathbf{F}T_s$ with T_s being the sampling period. The correlation matrix $\mathbf{Q}_1$ of the discrete-time white-noise process $\mathbf{w}_d$ can be found as $\mathbf{Q}_1 = \int_0^{T_s} \mathbf{\Phi}(\tau)\mathbf{Q}\mathbf{\Phi}^H(\tau)d\tau$. As such, $\mathbf{w}_d$ and $\mathbf{v}$ are assumed zero-mean Gaussian distributed white-noise processes, which are characterized by $\mathbf{Q}_1$ and $\mathbf{Q}_2$, the process and measurement noise matrices, respectively. To linearize the measurement equations, a first-order approximation is used in the continuous Riccati equations for the measurement matrix $\mathbf{H}(n)$ of dimension $N(N-1) \times 2DN$, which is the Jacobian matrix of $\mathbf{h}(\mathbf{x}(n))$ at the time instant of the most recent state estimate as

$$\mathbf{H}(n) = \nabla \mathbf{h}(\mathbf{x}(n))|_{\mathbf{x}=\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})} \tag{20}$$

where $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ denotes the estimate of $\mathbf{x}(n)$ given all the past measurements up to $y(n-1)$. Let $\mathbf{h}_{1,ij}(\mathbf{x}) = p_{ij} - u_{ij}$ and $\mathbf{h}_{2,ij}(\mathbf{x}) = s_{ij} - v_{ij}$, we have

$$\begin{aligned}
\frac{\partial \mathbf{h}_{1,ij}(\mathbf{x})}{\partial \mathbf{x}_i} &= -10 n_p (\log_{10} e) \frac{(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||^2} && (21)\\
\frac{\partial \mathbf{h}_{1,ij}(\mathbf{x})}{\partial \mathbf{x}_j} &= 10 n_p (\log_{10} e) \frac{(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||^2} && (22)\\
\frac{\partial \mathbf{h}_{1,ij}(\mathbf{x})}{\partial \dot{\mathbf{x}}_i} &= \mathbf{0}_{1\times D} && (23)\\
\frac{\partial \mathbf{h}_{1,ij}(\mathbf{x})}{\partial \dot{\mathbf{x}}_j} &= \mathbf{0}_{1\times D} && (24)
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \mathbf{h}_{2,ij}(\mathbf{x})}{\partial \mathbf{x}_i} &= -\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||^3} \\
&\quad + \frac{(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||} && (25)\\
\frac{\partial \mathbf{h}_{2,ij}(\mathbf{x})}{\partial \mathbf{x}_j} &= \frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||^3} \\
&\quad - \frac{(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||} && (26)
\end{aligned}$$

$$\underbrace{\begin{bmatrix}
0 & -\hat{\mathbf{x}}_{13}^T & 0 & & \cdots & 0 \\
0 & 0 & -\hat{\mathbf{x}}_{14}^T & 0 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & & \cdots & & 0 & -\hat{\mathbf{x}}_{1N}^T \\
\hat{x}_{23y} & -\hat{\mathbf{x}}_{23}^T & 0 & & \cdots & 0 \\
\hat{x}_{24y} & 0 & -\hat{\mathbf{x}}_{24}^T & 0 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & \cdots & & 0 & \hat{\mathbf{x}}_{(N-1)N}^T & -\hat{\mathbf{x}}_{(N-1)N}^T
\end{bmatrix}}_{\mathbf{R}'}
\underbrace{\begin{bmatrix}
\dot{x}_{2y} \\ \dot{\mathbf{x}}_3 \\ \vdots \\ \dot{\mathbf{x}}_{N-1} \\ \dot{\mathbf{x}}_N
\end{bmatrix}}_{\mathbf{v}'}
=
\underbrace{\begin{bmatrix}
s_{13} - \hat{\mathbf{x}}_{13}^T \dot{\mathbf{x}}_1 \\
s_{14} - \hat{\mathbf{x}}_{14}^T \dot{\mathbf{x}}_1 \\
\vdots \\
s_{1N} - \hat{\mathbf{x}}_{1N}^T \dot{\mathbf{x}}_1 \\
s_{23} - \hat{x}_{23x} \dot{x}_{2x} \\
s_{24} - \hat{x}_{24x} \dot{x}_{2x} \\
\vdots \\
s_{(N-1)N}
\end{bmatrix}}_{\mathbf{s}'}
\tag{12}$$

$$\frac{\partial \mathbf{h}_{2,ij}(\mathbf{x})}{\partial \dot{\mathbf{x}}_i} = \frac{(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||} \quad (27)$$

$$\frac{\partial \mathbf{h}_{2,ij}(\mathbf{x})}{\partial \dot{\mathbf{x}}_j} = -\frac{(\mathbf{x}_i - \mathbf{x}_j)^T}{||\mathbf{x}_i - \mathbf{x}_j||}. \quad (28)$$

The EKF equations are given by

$$\begin{aligned} \mathbf{G}(n) &= \mathbf{K}(n,n-1)\mathbf{H}^H(n)[\mathbf{H}(n)\mathbf{K}(n,n-1) \\ &\quad \mathbf{H}^H(n) + \mathbf{Q}_2(n)]^{-1} \quad (29) \\ \hat{\mathbf{x}}(n|\mathcal{Y}_n) &= \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}(n)[\mathbf{y}(n) \\ &\quad -\mathbf{h}(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))] \quad (30) \\ \hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{\Phi}\hat{\mathbf{x}}(n|\mathcal{Y}_n) \quad (31) \\ \mathbf{K}(n) &= [\mathbf{I} - \mathbf{G}(n)\mathbf{H}(n)]\mathbf{K}(n,n-1) \quad (32) \\ \mathbf{K}(n+1,n) &= \mathbf{\Phi}\mathbf{K}(n)\mathbf{\Phi}^H + \mathbf{Q}_1(n) \quad (33) \end{aligned}$$

with initial conditions

$$\begin{aligned} \hat{\mathbf{x}}(1|\mathcal{Y}_0) &= E[\mathbf{x}(1)] \quad (34) \\ \mathbf{K}(1,0) &= E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H]. (35) \end{aligned}$$

$\mathbf{G}(n)$ is the Kalman gain, $\mathbf{K}(n)$ is the filtered state-error correlation matrix, and the predicted state-error correlation matrix $\mathbf{K}(n+1,n)$ is computed recursively using the Riccati difference equation (33).

2) Distributed Extended Kalman Filter Approach: It makes sense to use distributed algorithm if the pair-wise measurements are only available between neighboring nodes and the communication overhead to the network is limited. Using the classic Matrix Inversion Lemma, i.e.,

$$(\mathbf{A} + \mathbf{BRC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{R}^{-1} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1} \quad (36)$$

where $\mathbf{A}$ is a nonsingular $n \times n$ matrix, $\mathbf{B}$ $n \times m$ matrix, $\mathbf{R}$ nonsingular $m \times m$ matrix, and $\mathbf{C}$ $m \times n$ matrix, from (29) and (32) we can show that the Kalman gain $\mathbf{G}(n)$ is given by

$$\mathbf{G}(n) = \mathbf{K}(n)\mathbf{H}^H(n)\mathbf{Q}_2^{-1}(n) \quad (37)$$

and the filtered state-error correlation matrix $\mathbf{K}(n)$ is given by

$$\mathbf{K}(n) = (\mathbf{K}^{-1}(n,n-1) + \mathbf{H}^H(n)\mathbf{Q}_2^{-1}(n)\mathbf{H}(n))^{-1}. \quad (38)$$

Let the inverses of the filtered and predicted state-error correlation matrices, $\mathbf{K}(n)$ and $\mathbf{K}(n,n-1)$, be the information matrices, $\mathbf{P}(n)$ and $\mathbf{P}(n,n-1)$ respectively. The Kalman filter iterations in their information form can be expressed as (Information-Filtering Algorithm)

$$\begin{aligned} \hat{\mathbf{x}}(n|\mathcal{Y}_n) &= \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{K}(n)\mathbf{H}^H(n)\mathbf{Q}_2^{-1}(n) \\ &\quad [\mathbf{y}(n) - \mathbf{h}(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))] \quad (39) \\ &= \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{K}(n)\mathbf{H}^H(n)\mathbf{Q}_2^{-1}(n) \\ &\quad [\tilde{\mathbf{y}}(n) - \mathbf{H}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})] \\ \hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{\Phi}\hat{\mathbf{x}}(n|\mathcal{Y}_n) \quad (40) \\ \mathbf{K}(n) &= (\mathbf{K}^{-1}(n,n-1) + \mathbf{H}^H(n)\mathbf{Q}_2^{-1}(n) \\ &\quad \mathbf{H}(n))^{-1} \quad (41) \\ \mathbf{K}(n+1,n) &= \mathbf{\Phi}\mathbf{K}(n)\mathbf{\Phi}^H + \mathbf{Q}_1(n) \quad (42) \end{aligned}$$

where $\tilde{\mathbf{y}}(n) = \mathbf{y}(n) - \mathbf{h}(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) + \mathbf{H}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$.

It is usually impractical to collect all-to-all node range and relative speed measurements. Suppose that the ith node is only sparsely connected with its neighboring nodes $j \in \mathcal{B}(i)$, where $\mathcal{B}(i)$ is set of neighboring nodes that connected with node i. The local measurement at node i is

$$\mathbf{y}_i(n) = \begin{bmatrix} \mathrm{col}(\{p_{ij}\}_{j\in\mathcal{B}(i)}) \\ \mathrm{col}(\{s_{ij}\}_{j\in\mathcal{B}(i)}) \end{bmatrix} \quad (43)$$

where $\mathrm{col}(\cdot)$ denotes the column extension. Suppose that the state estimates $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ are propagated through the network. (Actually, the estimates propagation through each neighborhood is suffice.) $\tilde{\mathbf{y}}(n)$ can be calculated locally as

$$\tilde{\mathbf{y}}_i(n) = \mathbf{y}_i(n) - \mathbf{h}_i(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) + \mathbf{H}_i(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \quad (44)$$

where

$$\mathbf{h}_i(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) = \begin{bmatrix} \mathrm{col}(\mathbf{h}_{1,ij\in\mathcal{B}(i)}(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))) \\ \mathrm{col}(\mathbf{h}_{2,ij\in\mathcal{B}(i)}(\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))) \end{bmatrix} \quad (45)$$

and

$$\mathbf{H}_i(n) = \begin{bmatrix} \mathrm{col}(\frac{\partial \mathbf{h}_{1,ij}(\mathbf{x})|_{j\in\mathcal{B}(i)}}{\partial \mathbf{x}}) \\ \mathrm{col}(\frac{\partial \mathbf{h}_{2,ij}(\mathbf{x})|_{j\in\mathcal{B}(i)}}{\partial \mathbf{x}}) \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})}. \quad (46)$$

With a diagonal matrix $\mathbf{Q}_{2i}$ corresponds to the correlation matrix associated with the local measurements, the local observation variables can be defined as

$$\begin{aligned} \mathbf{z}_i(n) &= \mathbf{H}_i^H(n)\mathbf{Q}_{2i}^{-1}(n)\tilde{\mathbf{y}}_i(n) \quad (47) \\ \mathbf{S}_i(n) &= \mathbf{H}_i^H(n)\mathbf{Q}_{2i}^{-1}(n)\mathbf{H}_i(n). \quad (48) \end{aligned}$$

Therefore the centralized information filter of the Kalman filter can be implemented by measurement fusion at a central location as

$$\begin{aligned} \hat{\mathbf{x}}(n|\mathcal{Y}_n) &= \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{K}(n)\left(\sum_{i=1}^{N}\mathbf{z}_i(n) \right. \\ &\quad \left. - \sum_{i=1}^{N}\mathbf{S}_i(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})\right) \quad (49) \\ \hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{\Phi}\hat{\mathbf{x}}(n|\mathcal{Y}_n) \quad (50) \\ \mathbf{K}(n) &= \left(\mathbf{K}^{-1}(n,n-1) + \sum_{i=1}^{N}\mathbf{S}_i(n)\right)^{-1} \quad (51) \\ \mathbf{K}(n+1,n) &= \mathbf{\Phi}\mathbf{K}(n)\mathbf{\Phi}^H + \mathbf{Q}_1(n). \quad (52) \end{aligned}$$

If the process is time-invariant, the $\{\mathbf{S}_i\}$ can be calculated off-line and stored locally. Using the information matrices, the Kalman filter iterations can be given by

$$\begin{aligned} \hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{\Phi}\mathbf{P}^{-1}(n)\left(\mathbf{P}(n,n-1)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \right. \\ &\quad \left. + \sum_{i=1}^{N}\mathbf{z}_i(n)\right) \quad (53) \\ \mathbf{P}(n) &= \mathbf{P}(n,n-1) + \sum_{i=1}^{N}\mathbf{S}_i(n) \quad (54) \\ \mathbf{P}^{-1}(n+1,n) &= \mathbf{\Phi}\mathbf{P}^{-1}(n)\mathbf{\Phi}^H + \mathbf{Q}_1(n) \quad (55) \end{aligned}$$

with initial conditions

$$\hat{\mathbf{x}}(1|\mathcal{Y}_0) = E[\mathbf{x}(1)] \quad (56)$$
$$\mathbf{P}^{-1}(1,0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H]. \quad (57)$$

It is clear that, when a central processing unit is available, the measurement fusion is simply $\mathbf{z}(n) = \sum_{i=1}^{N} \mathbf{z}_i(n)$ and $\mathbf{S}(n) = \sum_{i=1}^{N} \mathbf{S}_i(n)$. If appropriate local $\mathbf{P}$ and $\mathbf{\Phi}$ can be found and the Kalman filter be decomposed into distributed Kalman filters for each $\mathbf{z}_i(n)$ and $\mathbf{S}_i(n)$, not only does it reduce the network traffic by transferring less Kalman filter coefficients, but the computation is saved as well by inverting matrices of dimension $2D(|\mathcal{B}(i)|+1) \times 2D(|\mathcal{B}(i)|+1)$ instead of $2DN \times 2DN$. Here, $|\cdot|$ denotes the cardinality of the set. Of course, the savings in computation only occur when the network connection is sparse such that $|\mathcal{B}(i)| << N, \forall i$.

It should be mentioned that if $\mathbf{Q}_2$ is time-invariant, using information filtering algorithm the matrix inversion is on dimension $2DN \times 2DN$, whereas for the centralized algorithm using regular Kalman filter iterations, the matrix inversion is on dimension $N(N-1) \times N(N-1)$ for all-to-all connections. It depends on the connectivity (the measurement amount) and the size of the network to determine which Kalman algorithm to use for computational efficiency.

The measurements collected at node i are functions of the locations and velocities of the ith node itself and the nodes in its neighbor set $\mathcal{B}(i)$. Define the local state vector that is associated with node i as

$$\tilde{\mathbf{x}}_i(n) = \begin{bmatrix} \mathbf{x}_i(n) \\ \mathrm{col}(\mathbf{x}_j(n)|_{j \in \mathcal{B}(i)}) \\ \dot{\mathbf{x}}_i(n) \\ \mathrm{col}(\dot{\mathbf{x}}_j(n)|_{j \in \mathcal{B}(i)}) \end{bmatrix}. \quad (58)$$

The length of vector $\tilde{\mathbf{x}}_i$ is $N_i = 2(|\mathcal{B}(i)|+1)$. The distributed state-space model and measurement equations are given by

$$\tilde{\mathbf{x}}_i(n+1) = \mathbf{\Phi}_i \tilde{\mathbf{x}}_i(n) + \mathbf{w}_i(n) \quad (59)$$
$$\tilde{\mathbf{y}}_i(n) = \tilde{\mathbf{H}}_i(n)\tilde{\mathbf{x}}_i(n) + \mathbf{v}_i(n), \; i = 1, 2, \ldots, N \quad (60)$$

where $\tilde{\mathbf{y}}_i(n)$ is as defined before, $\tilde{\mathbf{H}}_i(n)$ is a sub-matrix of $\mathbf{H}_i(n)$ that takes the ith and jth ($j \in \mathcal{B}(i)$) columns of $\mathbf{H}_i(n)$. Given the structure of $\mathbf{\Phi}$, the state-space model of the cooperative localization and velocity estimation problem is inheritably distributed. $\mathbf{\Phi}_i$ is a sub-matrix of $\mathbf{\Phi}$ that takes the ith and jth ($j \in \mathcal{B}(i)$) columns of $\mathbf{\Phi}$. (59) and (60) lead to a set of N_i-dimensional local Kalman filters. However, the local state vectors $\{\tilde{\mathbf{x}}_i\}$ overlap with each other. Data fusion is required. Measurements corresponding to the shared states at multiple nodes should be fused. [1]

The local information filter of dimension N_i is developed at node i. The data are fused by exchanging information among the neighbors. Along the line of [11], the local information filter consists of the initial condition, a local filter step (including observation fusion and distributed matrix inversion), and a local prediction step (including estimation fusion). The local observation variables are modified as

$$\tilde{\mathbf{z}}_i(n) = \tilde{\mathbf{H}}_i^H(n)\mathbf{Q}_{2i}^{-1}(n)\tilde{\mathbf{y}}_i(n) \quad (61)$$
$$\tilde{\mathbf{S}}_i(n) = \tilde{\mathbf{H}}_i^H(n)\mathbf{Q}_{2i}^{-1}(n)\tilde{\mathbf{H}}_i(n) \quad (62)$$

where $\tilde{\mathbf{S}}_i(n)$ is a $2D(|\mathcal{B}(i)|+1) \times 2D(|\mathcal{B}(i)|+1)$ sub-matrix of $\mathbf{S}_i(n)$ that removes zero rows and columns.

IV. Simulation Results

We simulate a 2-D mobile ad-hoc network that has $N = 10$ nodes with unknown positions and velocities. Suppose that all pair-wise measurements are acquired. The range measurement is affected by the channel fading that is governed by the log-normal received power distribution as in (16).

First, we model the velocity such that the integration of the acceleration over a sampling period is a white Gaussian random variable. Statistically, this makes the variance of the velocity larger and larger over simulation iterations. This model fits the state-space model described by (15) along with its definition of parameters. With $M = 0$ to 3 reference nodes, Fig. 1 shows the network localization performance through the normalized root-mean-square-error (RMSE) in the position estimates. Fig. 2 shows the normalized RMSE in node velocity estimates, which are further normalized by their own average values at each iteration in order to make a fair demonstration under this velocity model.

Secondly, we model the velocity as white Gaussian random variables from one sampling period to the next, and that they have the same variance. $\mathbf{\Phi}$ in the discrete-time state-space model should be modified accordingly as

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_{DN \times DN} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \mathbf{F}T_s. \quad (63)$$

The simulation results of such model are shown in Fig. 3 and Fig. 4. Using the same Kalman iteration, the velocity estimates are always zeros. This is easy to see because the state-space model is reduced to only having position variables

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{w}_i(n). \quad (64)$$

It is shown that, when the node velocities $\{\dot{\mathbf{x}}\}$ are included in the state-space model, the network localization performance can be improved over the Kalman filter iterations. It is also revealed from the simulation results that the velocity is not tracked using the extended Kalman filtering method. In order to use the Kalman filter to track velocity, the acceleration $\ddot{\mathbf{x}}$ needs to be modeled and included in the state vector, and the measurement equation needs to be modified accordingly. Without $\ddot{\mathbf{x}}$ included in the state-space model, once the node positions are estimated, one can estimate the node velocities using methods described in Section III-B.

V. Conclusion

A cooperative network localization scheme is proposed for mobile ad-hoc networks where the pair-wise measurements are inaccurate. An extended Kalman filter that incorporates the node position estimation with the node velocity estimation

[1] When the mobile network evolves and the connectivity topology changes, the nodes need to be regrouped once in a while into neighborhoods with $\tilde{\mathbf{H}}_i$, $\mathbf{\Phi}_i$, $\tilde{\mathbf{x}}_i$, $\tilde{\mathbf{y}}_i$, etc. Regrouping of neighbors according to the changes in network connectivity is beyond the scope of this paper.

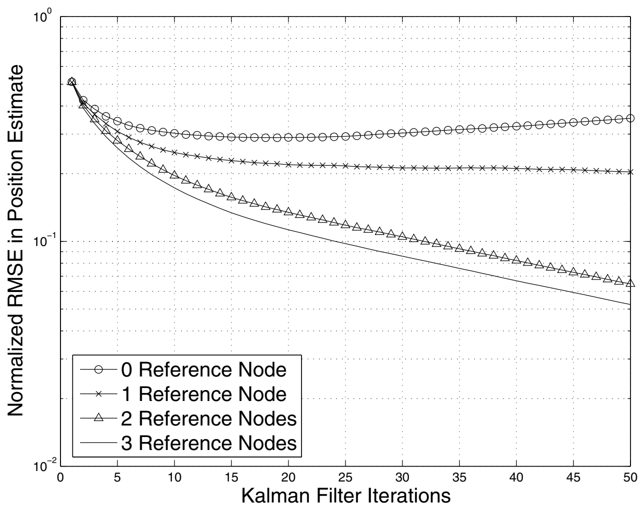

Fig. 1. Network localization performance. $N = 10$. Velocity model 1.

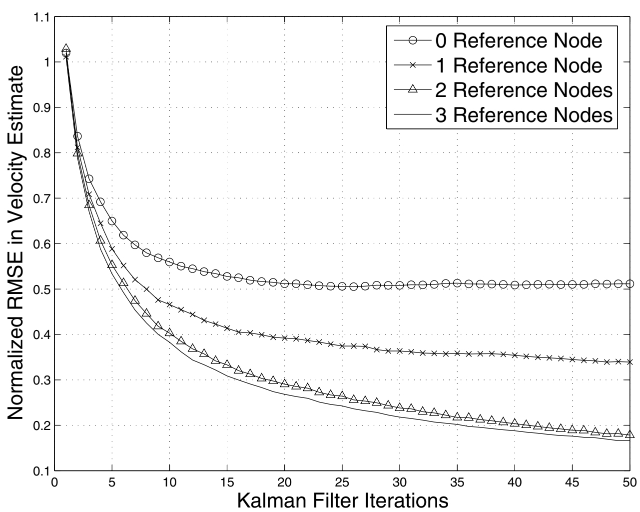

Fig. 2. Normalized node velocity estimation performance. $N = 10$. Velocity model 1.

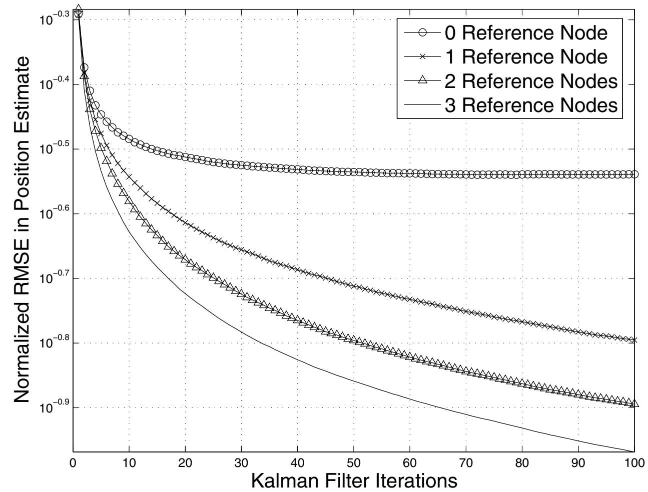

Fig. 3. Network localization performance. $N = 10$. Velocity model 2.

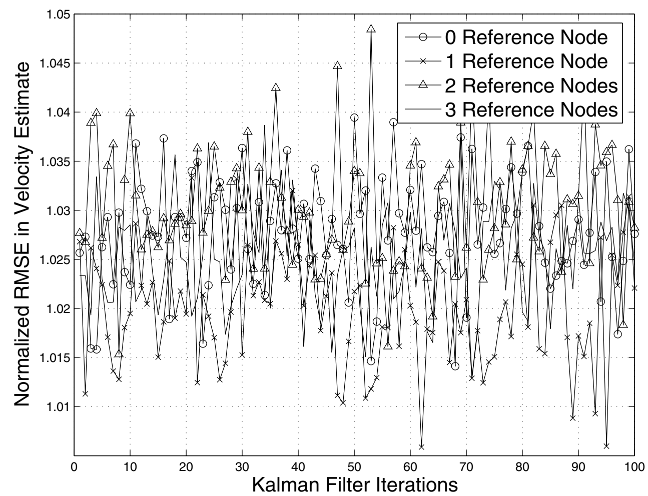

Fig. 4. Node velocity estimation performance. $N = 10$. Velocity model 2.

is used to improve the network localization performance. The algorithm is developed in a distributed manner for practical networks.

REFERENCES

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[2] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd ACM SenSys*, Nov. 2004.

[3] A. Basu, J. Gao, J. S. B. Mitchells, and G. Sabhnani, "Distributed localization using noisy distance and angle information," in *Proc. MobiHoc'06*, May 2006.

[4] C. Wang and L. Xiao, "Sensor localization under limited measurement capabilities," *IEEE Network*, vol. 21, no. 3, pp. 16–23, May/June 2007.

[5] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[6] G. Latsoudas and N. D. Sidiropoulos, "A fast and effective multidimensional scaling approach for node localization in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5121–5127, Oct. 2007.

[7] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[8] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

[9] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with GPS in urban canyon environments," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 15–25, Feb. 2003.

[10] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. IEEE Conf. Decision and Control and European Control Conf.*, Dec. 2005, pp. 8197–8184.

[11] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[12] A. J. Coulson, A. G. Williamson, and R. G. Vaughan, "A statistical basis for lognormal shadowing effects in multipath fading channels." *IEEE Trans. Veh. Technol.*, vol. 46, no. 4, pp. 494–502, Apr. 1998.