# DISTRIBUTED NONLINEAR KALMAN FILTERING WITH APPLICATIONS TO WIRELESS LOCALIZATION

*Federico S. Cattivelli* *Ali H. Sayed*

Department of Electrical Engineering
University of California, Los Angeles

## ABSTRACT

We study the problem of distributed state-space estimation, where a set of nodes are required to estimate the state of a nonlinear state-space system based on their observations. We extend our previous work on distributed Kalman filtering to the nonlinear case, and propose algorithms for Extended and Unscented Kalman filtering. The resulting algorithms are robust to node and link failure, scalable, and fully distributed, in the sense that no fusion center is required, and nodes communicate with their neighbors only. We apply the algorithms to the problem of estimating the position of every node in an ad-hoc network, also known as wireless localization. Simulation results illustrate the performance of the proposed algorithms.

***Index Terms***— Distributed estimation, distributed Kalman filtering, diffusion, adaptive networks, wireless localization.

## 1. INTRODUCTION

We study the problem of distributed estimation, where a set of nodes are required to collectively estimate the state of a nonlinear dynamical system based on their measurements. In a centralized solution to the problem, all nodes send their measurements to a fusion center, which uses a conventional filtering technique such as an Extended Kalman Filter (EKF) or an Unscented Kalman Filter (UKF) to obtain the *global* state estimate, and then sends the result to every node. This strategy may require large amounts of energy for communications and has a potential critical failure point at the fusion center.

Distributed implementations, on the other hand, avoid the use of a fusion center and distribute the processing and communication across the network. Among distributed processing algorithms, *diffusion algorithms* are amenable for real-time implementations, robust to node and link failure, scale with the size of the network, and obtain good performance in terms of estimation accuracy. Algorithms for diffusion LMS [1, 2], diffusion RLS [3] and diffusion Kalman filtering [4, 5, 6] have been proposed. Estimation algorithms based on average consensus have been proposed in [7, 8, 9, 10].

In this work we extend our previous work on linear distributed Kalman filtering [4, 6] to the nonlinear case. We propose two algorithms for nonlinear Kalman filtering, denoted diffusion EKF and diffusion UKF. These algorithms are diffusion-based, and therefore fully distributed, in the sense that no fusion center is required and estimates are obtained through local exchanges with neighbors only.

## 2. PROBLEM FORMULATION AND BACKGROUND

Consider a set of $N$ nodes distributed geographically over some region. We say that two nodes are connected if they can communicate directly with each other. The number of nodes connected to a certain node $k$ (including itself) is called the neighborhood of node $k$, and is denoted by $\mathcal{N}_k$. It is assumed that at every time instant $i$, every node $k$ in the network measures a vector $y_{k,i}$ which is related to an unknown state vector $x_i$ through the following state-space model:

$$\begin{cases} x_{i+1} &= f_i(x_i) + G_i n_i \\ y_{k,i} &= h_{k,i}(x_i) + v_{k,i} \end{cases} \quad (1)$$

where $x_i \in \mathbb{C}^M$, $y_{k,i} \in \mathbb{C}^{L_k}$ and $M$ and $L_k$ are positive integers for all $k$. The functions $f_i(\cdot)$ and $h_{k,i}(\cdot)$ are possibly nonlinear, and $G_i$ is a matrix. The signals $n_i$ and $v_{k,i}$ denote state and measurement noises, respectively, and are assumed to be zero-mean, and independent in time and space, with covariance matrices given by:

$$\mathrm{E}\, n_i n_j^* = Q_i \delta_{ij} \qquad \mathrm{E}\, v_{k,i} v_{l,j}^* = R_{k,i} \delta_{kl} \delta_{ij}. \quad (2)$$

The operator $^*$ denotes complex conjugate transposition and $\delta_{kl}$ is the Kronecker delta. The initial state $x_0$ is assumed to have mean $\mathrm{E}\, x_0$, with covariance matrix $\Pi_0 > 0$, and is uncorrelated with $n_i$ and $v_{k,i}$, for all $i$ and $k$. We further assume that $R_{k,i} > 0$.

The objective is for every node in the network to obtain an estimate of the current state $x_i$, given its observations and by collaborating with other nodes. We will denote by $\hat{x}_{k,i|j}$ the estimate of $x_i$ obtained by node $k$ given observations up to time $j$. Every node seeks to minimize the mean-square error (MSE) $\mathrm{E}\, \|x_i - \hat{x}_{k,i|i}\|^2$.

When model (1) is linear (i.e., $f_i(x) = F_i x$ and $h_{k,i}(x) = H_{k,i} x$), and the measurement noise, state noise, and initial state are jointly Gaussian, the optimal MSE estimate can be obtained by using a conventional Kalman filter. When model (1) is nonlinear, several approaches exist to deal with the problem of sequential state-space estimation. Two popular algorithms are the EKF and UKF [11, 12]. The EKF linearizes (1), and then applies a conventional Kalman filter to the resulting model. One difficulty with this algorithm is that it requires computing the derivatives of the state and measurement functions. On the other hand, the UKF does not have this requirement, and has been shown in some cases to outperform the EKF [12]. Distributed implementations of these two approaches, based on the diffusion strategy of [6], are presented in Section 3.

### 2.1. Diffusion Kalman filtering

The diffusion Kalman filter [4] is a fully distributed algorithm for state-space estimation in linear models. The diffusion KF algorithm and its variants require the introduction of a *diffusion* matrix $C \in \mathbb{R}^{N \times N}$ with the following properties:

$$\mathbb{1}^T C = \mathbb{1}^T \qquad c_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \qquad c_{l,k} \geq 0 \quad (3)$$

where $\mathbb{1}$ is an $N \times 1$ column vector with unit entries, and $c_{l,k}$ is the $(l,k)$ element of matrix $C$. The entries in $C$ represent the weights that are used by the diffusion algorithm to combine neighborhood estimates as follows:

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k} \psi_{l,i} \quad (4)$$

This work was supported in part by NSF grants ECS-0601266, ECS-0725441 and CCF-094936. Author's emails: {fcattiv, sayed}@ee.ucla.edu.



Authorized licensed use limited to: Univ of Calif Los Angeles. Downloaded on July 05,2010 at 16:11:07 UTC from IEEE Xplore. Restrictions apply.

where $\psi_{l,i}$ is obtained by applying conventional Kalman filtering recursions on the data from the neighborhood of node $l$ (see [6] for details). The diffusion step is an attempt to approximate the global KF performance via local node interactions. The algorithm (in information form) is summarized below for convenience.

**Algorithm 1: Diffusion Kalman filter [6]**
Consider a linear state-space model and a diffusion matrix as in (3). Start with $\hat{x}_{k,0|-1} = \mathrm{E}\, x_0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i$, compute at every node $k$:

**Step 1:** Measurement (incremental) update:

$$P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \sum_{l\in\mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} H_{l,i}$$

$$\psi_{k,i} = \hat{x}_{k,i|i-1} + P_{k,i|i} \sum_{l\in\mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} [y_{l,i} - H_{l,i}\hat{x}_{k,i|i-1}]$$

**Step 2:** Diffusion update:

$$\hat{x}_{k,i|i} = \textstyle\sum_{l\in\mathcal{N}_k} c_{l,k}\psi_{l,i}$$

**Step 3:** Time update:

$$\hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i}$$
$$P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$$

Algorithm 1 requires that at every instant $i$, nodes communicate with their neighbors their measurement matrices $H_{k,i}$, the covariance matrices $R_{k,i}$, and the measurements $y_{k,i}$. Subsequently, every node $k$ performs a measurement update to obtain $\psi_{k,i}$, and then communicates this intermediate estimate to its neighbors. Finally, every node $k$ combines the intermediate estimates $\psi_{l,i}$ from its neighbors in a convex manner, and performs a time-update. It is important to note that even though the notation $P_{k,i|i}$ and $P_{k,i|i-1}$ has been retained for simplicity in Alg. 1, these matrices do *not* represent the true covariances of the state estimates $\hat{x}_{k,i|i}$ and $\hat{x}_{k,i|i-1}$. Exact expressions for these covariances are presented in [4, 6].

## 3. DISTRIBUTED NONLINEAR KALMAN FILTERING

### 3.1. Selective Diffusion

Before proceeding with the presentation of the distributed nonlinear Kalman filtering algorithms, we introduce a technique denoted *selective diffusion*, which will be important in the wireless localization application presented in Section 4. Thus, instead of a diffusion update of the form (4), we will use the more general version:

$$[\hat{x}_{k,i|i}]_m = \sum_{l\in\mathcal{N}_k} c_{l,k,m}[\psi_{l,i}]_m \tag{5}$$

where

$$\sum_{l\in\mathcal{N}_k} c_{l,k,m} = 1 \;\text{ for all } m = 1,\ldots,M, \text{ and } c_{l,k,m} = 0 \text{ if } l \notin \mathcal{N}_k \tag{6}$$

Hence, the $m^{\text{th}}$ entry of $\hat{x}_{k,i|i}$ is calculated through a convex combination of the $m^{\text{th}}$ entries of $\{\psi_{l,i}\}$, but *the set of weights used for the combination depends on the entry* $m$. This approach can be generalized to the case where $c_{l,k}$ are matrices instead of scalars.

### 3.2. Diffusion Extended Kalman Filtering

We now proceed to derive a diffusion-based EKF. The algorithm is obtained by linearizing model (1) around certain desirable points, and applying the diffusion Kalman filtering algorithm (Alg. 1). Thus, consider again model (1). Linearizing these equations around some point $\psi$, we have the first order approximations:

$$\left\{\begin{array}{rcl} x_{i+1} & \approx & f_i(\psi) + \bar{F}_i(\psi)\cdot(x_i - \psi) + G_i n_i \\ y_{k,i} & \approx & h_{k,i}(\psi) + \bar{H}_{k,i}(\psi)\cdot(x_i - \psi) + v_{k,i} \end{array}\right. \tag{7}$$

where we defined the matrices

$$\bar{F}_i(\psi) = \left.\frac{\partial f_i(x)}{\partial x}\right|_{x=\psi} \qquad \bar{H}_{k,i}(\psi) = \left.\frac{\partial h_{k,i}(x)}{\partial x}\right|_{x=\psi}$$

Introducing

$$\begin{array}{rcl} \bar{y}_{k,i}(\psi) & = & y_{k,i} - h_{k,i}(\psi) + \bar{H}_{k,i}(\psi)\cdot\psi \\ \bar{u}_{k,i}(\psi) & = & f_i(\psi) - \bar{F}_i(\psi)\cdot\psi \end{array}$$

we obtain a linear model of the form:

$$\left\{\begin{array}{rcl} x_{i+1} & \approx & \bar{F}_i(\psi)\cdot x_i + G_i n_i + \bar{u}_{k,i}(\psi) \\ \bar{y}_{k,i}(\psi) & \approx & \bar{H}_{k,i}(\psi) x_i + v_{k,i} \end{array}\right. \tag{8}$$

where $\bar{u}_{k,i}(\psi)$ is a deterministic input to the model. The above model clearly depends on the point $\psi$ used to evaluate the state and measurement equations, and this point should be as close as possible to the true state $x_i$ in order to have a good approximation. Notice from Alg. 1, that in order to compute the measurement update, we need to know $H_{k,i}$. The best estimate of $x_i$ known at this point is $\hat{x}_{k,i|i-1}$. Thus, we replace $\psi$ in the measurement equation of (8) with $\hat{x}_{k,i|i-1}$. During the diffusion update, we already have access to $\hat{x}_{k,i|i}$, and therefore this is the best estimate of $x_i$. Thus, we may replace $\psi$ in the state equation of (8) with $\hat{x}_{k,i|i}$. The complete proposed algorithm is shown below.

**Algorithm 2: Diffusion Extended Kalman filter**
Consider a state-space model as in (1) and a diffusion matrix as in (6). Start with $\hat{x}_{k,0|-1} = \mathrm{E}\, x_0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i \geq 0$, compute at every node $k$:

**Step 1:** Measurement (incremental) update:

$$\hat{H}_{k,l,i} = \bar{H}_{l,i}(\hat{x}_{k,i|i-1})$$

$$P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \sum_{l\in\mathcal{N}_k} \hat{H}_{k,l,i}^* R_{l,i}^{-1} \hat{H}_{k,l,i}$$

$$\psi_{k,i} = \hat{x}_{k,i|i-1} + P_{k,i|i} \sum_{l\in\mathcal{N}_k} \hat{H}_{k,l,i}^* R_{l,i}^{-1} [y_{l,i} - h_{l,i}(\hat{x}_{k,i|i-1})]$$

**Step 2:** Diffusion update:

$$[\hat{x}_{k,i|i}]_m \leftarrow \textstyle\sum_{l\in\mathcal{N}_k} c_{l,k,m}[\psi_{l,i}]_m \quad \text{(selective diffusion)}$$

**Step 3:** Time update:

$$\hat{x}_{k,i+1|i} = \bar{F}_i(\hat{x}_{k,i|i})\cdot\hat{x}_{k,i|i} + \bar{u}_{k,i}(\hat{x}_{k,i|i})$$
$$P_{k,i+1|i} = \bar{F}_i(\hat{x}_{k,i|i}) P_{k,i|i} \bar{F}_i(\hat{x}_{k,i|i})^* + G_i Q_i G_i^*$$

Notice that in addition to the measurements $y_{l,i}$ and covariances $R_{l,i}$, node $k$ also needs to have access to the measurement functions of its neighbors and their derivatives, namely $h_{l,i}(\cdot)$ and $\bar{H}_{l,i}(\cdot)$.

### 3.3. Diffusion Unscented Kalman Filter

We now derive an Unscented Kalman filtering (UKF) version of Algorithm 1. We proceed in a similar fashion as we did with the diffusion EKF (Alg. 2). Let $n_k$ denote the number of neighbors of node $k$ (including itself), and $k_1, k_2 \ldots, k_{n_k}$ denote the indexes of these neighbors (i.e., $\mathcal{N}_k = \{k_1, k_2, \ldots, k_{n_k}\}$). Since every node $k$ has access to the measurements of its neighbors, we can consider the



Authorized licensed use limited to: Univ of Calif Los Angeles. Downloaded on July 05,2010 at 16:11:07 UTC from IEEE Xplore. Restrictions apply.

following augmented observation for node $k$, and its corresponding measurement noise covariance matrix

$$\begin{aligned}\breve{y}_{k,i} &= \mathrm{col}\{y_{k_1}, y_{k_2}, \ldots, y_{n_k}\} = \breve{h}_{k,i}(x_i)\\ \breve{R}_{k,i} &= \mathrm{diag}\{R_{k_1,i}, R_{k_2,i}, \ldots, R_{n_k,i}\}\end{aligned}$$

where the operators $\mathrm{col}\{\cdot\}$ and $\mathrm{diag}\{\cdot\}$ work by stacking their arguments column- and diagonal-wise, respectively.

Assume a model as in (1), and let $\hat{x}_{k,i|i-1}$ and $\hat{y}_{k,i|i-1}$ denote the linear minimum MSE estimators of $x_i$ and $\breve{y}_{k,i}$, respectively, given observations $\{\breve{y}_{k,j}\}$ up to time $i-1$, and let $P_{k,i|i-1}$ denote the covariance matrix of $\hat{x}_{k,i|i-1}$. A well known result in linear estimation theory [13] states that given a new observation $\breve{y}_{k,i}$ at time $i$, we can update the optimal linear estimator as follows:

$$\begin{aligned}\hat{x}_{k,i|i} &= \hat{x}_{k,i|i-1} + K_{k,i}(y_{k,i} - \hat{y}_{k,i|i-1})\\ P_{k,i|i} &= P_{k,i|i-1} - K_{k,i}R_{e,k,i}K_{k,i}^*\\ K_{k,i} &= R_{xe,k,i}R_{e,k,i}^{-1}\end{aligned} \tag{9}$$

where $R_{xe,k,i} = \mathrm{E}(x_i e_{k,i}^*)$, $R_{e,k,i} = \mathrm{E}\, e_{k,i}e_{k,i}^*$ and $e_{k,i} = \breve{y}_{k,i} - \hat{y}_{k,i|i-1}$. We also have

$$\begin{aligned}\hat{y}_{k,i|i-1} &= \mathrm{E}[h(x_i)|\{\breve{y}_{k,j}\}_{j=0,\ldots,i-1}]\\ \hat{x}_{k,i+1|i} &= \mathrm{E}[f(x_i)|\{\breve{y}_{k,j}\}_{j=0,\ldots,i}]\end{aligned}$$

The UKF [11, 12] computes the above expectations by propagating sigma-points. Assume node $k$ has an estimate of $x_i$, $\hat{x}_{k,i|i-1}$, and an estimate of its covariance matrix, $P_{k,i|i-1}$. The first step is for node $k$ to draw a set of $2M+1$ sigma points as follows:

$$\mathcal{X}_{k,i}^- = \hat{x}_{k,i|i-1}\mathbb{1}^T + [0 \quad \gamma P_{k,i|i-1}^{1/2} \quad -\gamma P_{k,i|i-1}^{1/2}] \tag{10}$$

where $P_{k,i|i-1}^{1/2}$ is a lower triangular Cholesky factor of $P_{k,i|i-1}$ and $\mathbb{1}$ is a vector of dimension $2M+1$ with unit entries. Next, the sigma-points are propagated through the measurement equation:

$$\mathcal{Y}_{k,i} = \left[\breve{h}_{k,i}(\mathcal{X}_{k,i}^- e_1) \;\; \breve{h}_{k,i}(\mathcal{X}_{k,i}^- e_2) \;\; \ldots \;\; \breve{h}_{k,i}(\mathcal{X}_{k,i}^- e_{2M+1})\right] \tag{11}$$

where $e_L$ is a vector with a unit entry at position $L$ and zeros elsewhere. We now obtain approximations for $\hat{y}_{i|i-1}$, its covariance, and the cross-covariance with $x_i$, namely,

$$\begin{aligned}\hat{y}_{k,i|i-1} &= \mathcal{Y}_{k,i}w^{(m)}\\ R_{e,k,i} &= (\mathcal{Y}_{k,i} - \hat{y}_{k,i|i-1}\mathbb{1}^T)W^{(c)}(\mathcal{Y}_{k,i} - \hat{y}_{k,i|i-1}\mathbb{1}^T)^* + \breve{R}_{k,i}\\ R_{xe,k,i} &= (\mathcal{X}_{k,i}^- - \hat{x}_{k,i|i-1}\mathbb{1}^T)W^{(c)}(\mathcal{Y}_{k,i} - \hat{y}_{k,i|i-1}\mathbb{1}^T)^*\end{aligned} \tag{12}$$

where $w^{(m)}$ and $W^{(c)}$ are weighting parameters. The measurement update is now:

$$\begin{aligned}\psi_{k,i} &= \hat{x}_{k,i|i-1} + K_{k,i}(\breve{y}_{k,i} - \hat{y}_{k,i|i-1})\\ P_{k,i|i} &= P_{k,i|i-1} - K_{k,i}R_{e,k,i}K_{k,i}^*\\ K_{k,i} &= R_{xe,k,i}R_{e,k,i}^{-1}\end{aligned} \tag{13}$$

Compared to (9), notice that we are now using the intermediate variable $\psi_{k,i}$. As was the case for the EKF, the diffusion update is:

$$[\hat{x}_{k,i|i}]_m \leftarrow \sum_{l\in\mathcal{N}_k} c_{l,k,m}[\psi_{l,i}]_m \quad \text{(selective diffusion)} \tag{14}$$

In order to perform a time-update, we again draw $2M+1$ sigma-points around the latest estimate, propagate them through the state equation, and compute the new mean and covariance as follows:

$$\begin{aligned}\mathcal{X}_{k,i} &= \hat{x}_{k,i|i}\mathbb{1}^T + [0 \quad \gamma P_{k,i|i}^{1/2} \quad -\gamma P_{k,i|i}^{1/2}]\\ \mathcal{X}_{k,i+1}^\dagger &= [f_i(\mathcal{X}_{k,i}e_1) \;\; f_i(\mathcal{X}_{k,i}e_2) \;\; \ldots \;\; f_i(\mathcal{X}_{k,i}e_{2M+1})]\\ \hat{x}_{k,i+1|i} &= \mathcal{X}_{k,i+1}^\dagger w^{(m)}\\ P_{k,i+1|i} &= (\mathcal{X}_{k,i+1}^\dagger - \hat{x}_{k,i+1|i}\mathbb{1}^T)W^{(c)}(\mathcal{X}_{k,i+1}^- - \hat{x}_{k,i+1|i}\mathbb{1}^T)^*\end{aligned} \tag{15}$$

The algorithm parameters [12] are given by:

$$\begin{aligned}&\lambda = \alpha^2(M+\kappa) - M\\ &\gamma = \sqrt{M+\lambda}\\ &w^{(m)} = \left[\tfrac{\lambda}{M+\lambda} \;\; \tfrac{2}{M+\lambda} \;\; \tfrac{2}{M+\lambda}\cdots\tfrac{2}{M+\lambda}\right]^T\\ &W^{(c)} = \mathrm{diag}\left\{\tfrac{\lambda}{M+\lambda}+\delta, \tfrac{2}{M+\lambda}, \tfrac{2}{M+\lambda}, \ldots, \tfrac{2}{M+\lambda}\right\}\end{aligned} \tag{16}$$

Typical values of the parameters are $\alpha = 10^{-4}$, $\kappa = 3-M$, $\beta = 2$ and $\delta = 1-\alpha^2+\beta$. The complete algorithm is shown below.

**Algorithm 3: Diffusion Unscented Kalman filter**

Consider a state-space model as in (1) and a diffusion matrix as in (6). Start with $\hat{x}_{k,0|-1} = \mathrm{E}\, x_0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i \geq 0$, repeat at every node $k$:

- Calculate $2M+1$ sigma-points using (10) and propagate through the measurement equation using (11).
- Perform a measurement-update using (12) and (13).
- Perform a diffusion update using (14).
- Calculate $2M+1$ new sigma-points, propagate through the state equation and perform time-update using (15).

## 4. APPLICATION: WIRELESS LOCALIZATION

We now apply the proposed nonlinear estimation algorithms (Alg. 2 and 3) to the problem of wireless localization. Every node is required to obtain an estimate of its own position (assumed to be 2-dimensional) based on their measurements. It is also assumed that there is a set of *anchor* nodes that know their positions exactly. The initial network topology is shown in Fig. 1, with the anchors denoted as red squares. All nodes are placed in the $[0,1]\times[0,1]$ square. Let $\{k_1, k_2, \ldots, k_{n_k-1}\}$ denote the set of neighbors of node $k$, *excluding* itself. It is assumed that if nodes $k$ and $k_l \neq k$ are neighbors, and are located at time $i$ at positions $(x_{k,i}, y_{k,i})$, and $(x_{k_l,i}, y_{k_l,i})$, respectively, then node $k$ obtains a measurement of the form:

$$d_{k,l,i} = \eta\log\left(\epsilon + \sqrt{(x_{k,i}-x_{k_l,i})^2 + (y_{k,i}-y_{k_l,i})^2}\right)$$

At time $i$, the measurement vector obtained by node $k$ and the state vector of the network are given, respectively, by:

$$\begin{aligned}y_{k,i} &= [d_{k,1,i} \quad \ldots \quad d_{k,n_k-1,i}]^T + v_{k,i}\\ x_i &= [x_{1,i} \;\; y_{1,i} \;\; x_{2,i} \;\; y_{2,i} \;\; \ldots \;\; y_{N,1} \;\; y_{N,2}]^T\end{aligned}$$

In our simulations we use a linear state equation, with $f_i(x_i) = x_i$. We also use $\Pi_0 = 10^{-4}I_M$, $G = I$, $Q = 2\times10^{-5}I$, $R_{k,i} = 5\sqrt{k}\times10^{-4}I_{n_k-1}$, $\eta = -1$ and $\epsilon = 0.1$. The initial states are $\hat{x}_{k,0|-1} = 0.5\mathbb{1}$ for all $k$.

Notice that in our observation model, node $k$ only observes the state of its neighbors. Thus, any estimate of the position of any other node may be unreliable. For this reason, in the diffusion update of Algorithms 2 and 3, a node will combine in a convex manner the estimates of its neighbors, only for those entries that are reliable.



Authorized licensed use limited to: Univ of Calif Los Angeles. Downloaded on July 05,2010 at 16:11:07 UTC from IEEE Xplore. Restrictions apply.

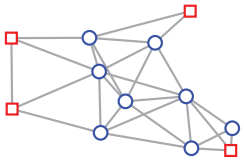

**Fig. 1**. Network topology, where red boxes are anchor nodes.

Thus, for the diffusion coefficients we use the following rule:

$$c_{l,k,m} = \begin{cases} \alpha_{k,m} n_l & \text{if } l \in \mathcal{N}_k \text{ and node } l \text{ observes state } m \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $\alpha_{k,m}$ is a normalizing constant such that (6) is satisfied. Every node is weighted proportionally to its degree. If neighbor $l$ of node $k$ does not observe state $m$, node $k$ uses a weight zero for this entry in the diffusion update.

Figure 2 shows the transient performance of the proposed algorithms, compared to other cooperation schemes. The performance is measured in terms of the estimation error obtained by each node when estimating its own position only (the estimates of the positions of other nodes are not included), averaged over all nodes and over 50 independent experiments. The global solution could be computed, for example, by a fusion center in a centralized strategy. The "no cooperation" algorithm corresponds to the case where nodes do not exchange measurements with their neighbors, and run a conventional EKF or UKF on their own data. The "local" algorithm corresponds to the case where nodes exchange their measurements with their neighbors, and run a conventional EKF or UKF on this data, which includes their own measurements and the measurements of the neighbors. Note that in this case no diffusion is present and intermediate estimates are not exchanged. The plots show that both the diffusion EKF and UKF algorithms have a performance which is close to the global solutions. In comparing the EKF and UKF solutions, we did not find a significant difference in our simulations. Fig. 3 shows the estimated node positions for one experiment, using the diffusion UKF algorithm (Alg. 3). Notice that the positions are correctly estimated after a few iterations.

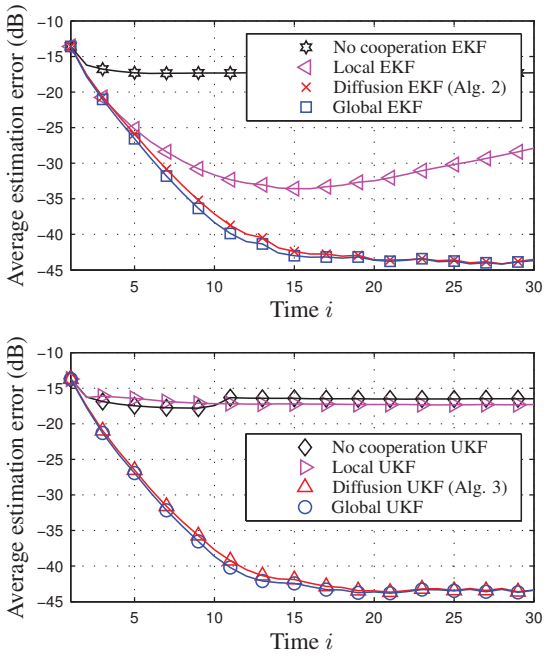


**Fig. 2**. Comparison of EKF (top) and UKF (bottom) strategies.

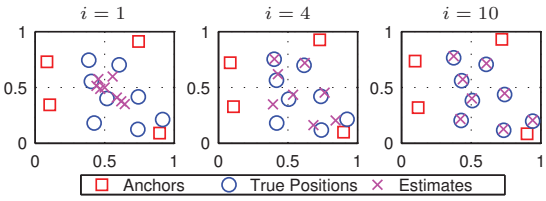


**Fig. 3**. Actual node positions and estimates at different time instants.

## 5. CONCLUSIONS

We proposed diffusion EKF and UKF algorithms for distributed estimation in non-linear dynamic systems. Our simulations indicate that the algorithms can achieve a performance close to the global solution for the problem of distributed wireless localization.

## 6. REFERENCES


[1] A. H. Sayed and C. G. Lopes, "Adaptive estimation algorithms over distributed networks," in *Proc. 21st IEICE Signal Processing Symposium*, Kyoto, Japan, November 2006.

[2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[3] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[4] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis," in *Proc. Cognitive Information Processing*, Santorini, Greece, June 2008.

[5] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," in *Proc. EUSIPCO*, Lausanne, Switzerland, August 2008.

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," to appear, *IEEE Trans Automatic Control*, 2010.

[7] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc IPSN*, Nashville, TN, April 2006, pp. 168–176.

[8] I. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links part II: distributed estimation and smoothing of random signals," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1650–1666, 2008.

[9] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decision and Control*, New Orleans, LA, December 2007.

[10] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Processing*, vol. 56, no. 10, Part 1, pp. 4919–4935, October 2008.

[11] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. American Control Conf.*, Seattle, Washington, June 1995, pp. 1628–1632.

[12] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in Kalman Filtering and Neural Networks, S. Haykin, editor, Wiley, NY, 2001.

[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, NJ, 2000.




Authorized licensed use limited to: Univ of Calif Los Angeles. Downloaded on July 05,2010 at 16:11:07 UTC from IEEE Xplore. Restrictions apply.