

Research Article

Neighborhood Kalman Estimation for Distributed Localization in Wireless Sensor Networks

Xiaochu Wang,^{1,2} Ting Sun,² and Chunshi Fan¹

¹Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing 100094, China

²Department of Precision Instrument, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Ting Sun; sunting09@tsinghua.org.cn and Chunshi Fan; fancs@139.com

Received 10 November 2015; Revised 7 January 2016; Accepted 26 January 2016

Academic Editor: Paolo Addresso

Copyright © 2016 Xiaochu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate location information plays an important role in the performance of wireless sensor networks since many mission applications depend on it. This paper proposes a fully distributed localization algorithm based on the concept of data fusion, allowing the full intranodes information including the correlations among estimates to take part in the algorithm. By properly constructing and updating the estimates as well as the corresponding covariance matrices, the algorithm can fuse intranodes information to generate more accurate estimates on the sensor locations with a fast speed. Finally, numerical simulations are given as examples to demonstrate the effectiveness of the algorithm.

1. Introduction

Wireless sensor networks (WSNs) have attracted considerable attention since large numbers of simple sensor nodes working cooperatively can achieve much more complex tasks, such as environmental monitoring, target tracking, and synthetic aperture. With recent advances in the rapid development of small-sized and low-cost smart sensor nodes, large-scale deployed WSNs have become viable and promising.

Sensor node localization is crucial in most of the above-mentioned cooperative tasks, because the location information plays an important role in the coordinated performance. Generally, localization techniques can be classified into centralized approaches and distributed approaches, and the latter approaches are believed to be more efficient and flexible in large-scale networks. Although the existing algorithms can give solutions to the localization problem, there is still a great potential to research more on it.

In this paper, we depend on the perspective of data fusion and propose a distributed localization algorithm, namely, the constructed neighborhood Kalman filter (CNKF). It is a fully distributed algorithm where each sensor node only cooperates with its neighboring sensor nodes to get the localization with noisy distance measurements. The algorithm is easily implementable for large-scale networks and can make full use of the intranetwork information.

The rest of this paper is organized as follows. Preliminaries are given in Section 2. Related works are reviewed in Section 3. The CNKF algorithm is proposed in detail in Section 4. Simulations are given and discussed in Section 5. Finally, conclusions are remarked in Section 6.

2. Preliminaries

2.1. The Kalman Filter. We will review in this section the principal equations for the extended Kalman filter in order to introduce the necessary notation for the following sections. The Kalman filter is a real-time estimation algorithm, which can estimate the “state” (state means the variable desired to be estimated) by predicting and correcting it at discrete moments.

The main preconditions of the Kalman filter are twofold. Firstly, the change of the state should be predictable. That is, an equation describing how the state is changing with time is necessary. Such an equation is named “state equation.” Secondly, the measurement and the state should be quantitatively related. That is, a functional equation between the measurement and the state is desired. Such an equation is named “measurement equation.”

At each moment, the summary of procedure of the Kalman filter is known as follows. (a) State prediction,

according to the state equation, predicts the theoretical value of the state based on the estimated value at the previous moment. (b) Measurement prediction, according to the measurement equation, predicts the theoretical value of the measurement based on the predicted value of the state. (c) Measurement observation is observing the actual value of the measurement. (d) State correction finds the difference between the predicted value and actual value of the measurement and compensates for the predicted value of the state. Since steps (b), (c), and (d) as a whole can refine the predicted value of the state, they can also be called “state filtering.”

The formulas of the Kalman filter are reviewed as follows, acting as a basic framework. These formulas were proposed under strict derivation, which can be found in [1] and hence detailed derivation is omitted in this paper.

Without loss of generality, assume that the state equation may be written as

$$X|_k = \Phi|_{k/k-1} X|_{k-1} + \Gamma|_{k/k-1} W|_{k-1}. \quad (1)$$

Similarly, assume that the measurement equation may be written as

$$\tilde{Z}|_k = h(X|_k) + V|_k, \quad (2)$$

where $|_{t=k}$ is the time stamp to distinguish variables at different moments, Φ and Γ are state-transition matrices, and X is the state vector. W and V (the process noise and measurement noise, resp.) are Gaussian white-noise processes whose mean and covariance function are given by

$$\begin{aligned} E[W] &= 0, \\ E[V] &= 0, \\ E[WW^T] &= Q, \\ E[VV^T] &= R, \end{aligned} \quad (3)$$

where symbol E denotes the expectation and T matrix transpose, respectively.

Assume that the initial guesses of the mean and covariance of the state vector at time $t = 0$ are given by

$$\begin{aligned} E[X|_0] &\equiv \hat{X}|_0, \\ E[(X|_0 - \hat{X}|_0)(X|_0 - \hat{X}|_0)^T] &\equiv P|_0. \end{aligned} \quad (4)$$

After being initialized, the Kalman filter will enter the iterative procedure.

State Prediction. At each moment k , the state at time $t = k$ can be predicted based on the estimated state at time $t = k - 1$ according to (1). The predicted state can be obtained as

$$\begin{aligned} \hat{X}|_{k/k-1} &= \Phi|_{k/k-1} \hat{X}|_{k-1}, \\ P|_{k/k-1} &= \Phi|_{k/k-1} P|_{k-1} \Phi|_{k/k-1}^T \\ &\quad + \Gamma|_{k/k-1} Q|_{k-1} \Gamma|_{k/k-1}^T. \end{aligned} \quad (5)$$

State Filtering. Once the state has been predicted, the theoretical measurement at time $t = k$ can be predicted according to (2)

$$\tilde{Z}|_{k/k-1} = h(\hat{X}|_{k/k-1}) \quad (6)$$

and the minimum variance estimate of the state immediately following the measurement is given by

$$\hat{X}|_k = \hat{X}|_{k/k-1} + K|_k (\tilde{Z}|_k - \tilde{Z}|_{k/k-1}), \quad (7)$$

where the Kalman gain matrix is given by

$$K|_k = P|_{k/k-1} H|_k^T (H|_k P|_{k/k-1} H|_k^T + R|_k)^{-1} \quad (8)$$

and the measurement sensitivity matrix is given by

$$H|_k = \left. \frac{\partial h(X)}{\partial X} \right|_{\hat{X}|_{k/k-1}}. \quad (9)$$

It is worthwhile to note that (7) is the optimal way to correct the predicted state, so that the estimated state can achieve the minimum variance. Following the estimated state, the covariance matrix of the estimate is given by

$$\begin{aligned} P|_k &= (I - K|_k H|_k) P|_{k/k-1} (I - K|_k H|_k)^T \\ &\quad + K|_k R|_k K|_k^T. \end{aligned} \quad (10)$$

The above is the formulaic framework of the Kalman filter. One may also see some more literature for closer understanding on how it works in many applications (see, e.g., [2, 3]).

2.2. The Covariance Intersection Fusion. The necessity and principal equations of the fusion of correlated estimates will be reviewed in this subsection.

When following the Kalman filtering scheme, the estimates to be fused must satisfy the assumption that they are independent. However, in many conditions where estimates are highly correlated but the degree of the correlation is not known exactly, applying the Kalman filter might lead to overconfidence in the accuracy of the fused estimate and result in divergent iterations. In this case, the covariance intersection algorithm is suitable for providing a conservative confidence fusion for these correlated estimates [4].

Given two estimates $\{\hat{X}_a, P_a\}$ and $\{\hat{X}_b, P_b\}$, the covariance intersection algorithm can be summarized as

$$\begin{aligned} P_c^{-1} &= \omega P_a^{-1} + (1 - \omega) P_b^{-1}, \\ P_c^{-1} \hat{X}_c &= \omega P_a^{-1} \hat{X}_a + (1 - \omega) P_b^{-1} \hat{X}_b, \end{aligned} \quad (11)$$

where $\omega \in [0, 1]$ is the weight. The weight can be further optimized with respect to different performance criteria such as minimising the trace or the determinant of P_c .

2.3. Definitions. In this paper, all the sensor nodes in the network being studied are assumed to be equipped with the same wireless communication modules and computation modules. Sensor nodes are divided into two types, the ordinary nodes and the anchor nodes.

Definition 1 (ordinary nodes). The ordinary nodes refer to the sensor nodes that do not know their locations. We denote by \mathcal{O} the set of the ordinary nodes in the network.

Definition 2 (anchor nodes). The anchor nodes refer to the sensor nodes that are aware of their locations with respect to a given coordinate frame. We denote by \mathcal{A} the set of the ordinary nodes in the network.

Definition 3 (neighbors). If two sensor nodes can communicate with each other in short distances, they are called neighbors. It is assumed that the distance information between a pair of neighbors can be measured. If the actual distance between nodes i and j is d_{ij} , then the distance measurement is given by

$$\tilde{d}_{ij} = d_{ij} + \varepsilon_{ij}, \quad (12)$$

where ε_{ij} is the measurement noise (see, e.g., [5]) following $\varepsilon_{ij} \sim N(0, \sigma_{\varepsilon_{ij}}^2)$ and $\sigma_{\varepsilon_{ij}}$ is influenced by the distance and the noise ratio Y , simulated by

$$\sigma_{\varepsilon_{ij}} = d_{ij} * \frac{Y}{100}. \quad (13)$$

Problem Statement. Suppose that there exists a static WSN consisting of n ordinary nodes and m anchor nodes, and the neighboring distance measurements among the nodes are available but noisy. After giving an initial guess on the localization as the starting condition, the problem is how to achieve a much better localization fast and accurately.

3. Related Works

General overviews of localization approaches for WSNs can be found in lots of literatures (see, e.g., [6–10]). Localization approaches can be simply classified into the centralized type and the distributed type, and the latter is widely believed to be more efficient than the former for large-scale networks because communication with only the neighboring sensor nodes is much easier than with all the sensor nodes in the network [11]. In distributed approaches, incremental algorithms and concurrent algorithms are two fundamental thoughts. Incremental algorithms determine the location information node by node increasingly. That is, in each moment, the incremental algorithms start with three (or more) neighboring anchor nodes to solve the location information of a fourth node that is located inside the overlapped area, and then the fourth node will serve as a new anchor node in the next moment as soon as it knows its location (see, e.g., [12–14]). The main drawback of the incremental algorithms is that errors may accumulate and increase as the propagation spreading [8, 15]. Some assistant strategies can be adopted in order to improve the localization accuracy of the incremental algorithms, such as using more stable anchor nodes [16] and utilizing the local geometrical constraint [17]. Unlike incremental algorithms, concurrent algorithms determine the location information of all the sensor nodes simultaneously, where all the sensor nodes behave indistinctively and update the location information in parallel without

interference, and then the global localization result of the whole network will be achieved asymptotically by neighboring cooperation as the time goes on (see, e.g., [15, 18–20]). Among concurrent algorithms, the mass-spring relaxation is the most popular algorithm, which treats location estimates and distance measurements as masses and springs, respectively, in an imaginary mass-spring system. By relaxing the masses and letting them move under spring forces, the final location estimates will be gotten when the system achieves the equilibrium [15]. There are also some algorithms considering the statistical characteristics such as [21–23] from which it can be seen that the location estimates can be further refined by statistical approaches under certain conditions such as linear assumptions and a priori knowledge. Recently, a so-called distributed Kalman filter has been studied (see, e.g., [24, 25]), which focuses on the problem of overlapping estimation. Overlapping estimation means that different separate objects are interested in an overlapped common state and estimate on it cooperatively. The distributed Kalman filter requires all the objects (i.e., sensor nodes) to be able to observe the overlapped common state, and therefore, it cannot be applied in the localization problem. Nevertheless, the thought of the distributed Kalman filter brings much inspiration to researchers.

4. Constructed Neighborhood Kalman Filter

We will propose the constructed neighborhood Kalman filter for distributed localization in this section. Overall speaking, the main idea of the CNKF can be summarized into four points. (a) Construct a proper form of the state containing not only the ego's location but also the neighbors' locations. By doing this, the states of any two neighboring ordinary nodes would contain an overlapped part, and this provides each ordinary node with the feasibility of cooperating with the neighbors. (b) Use the static nature of static sensor networks well. Note that the state constructed by each ordinary node contains the neighbors' locations, so the state would be very hard to predict if the neighbors were able to move. Fortunately, the nodes in static sensor networks are known to be static, or in other words, the change of the state should be zero. Therefore, the state equation (14) is suitable for predicting the state. (c) Use the Kalman filter to locally refine the state, and use the covariance intersection algorithm to cooperatively refine the overlapped part of the state. (d) Reconstruct the state by neighboring cooperation properly, and reconstruct the covariance approximately satisfying both the positive definiteness (in order to guarantee the consecutive iterations) and the near-minimum variance (in order to achieve the optimal estimation). In detail, the procedures and formulas of the CNKF are shown as follows.

State Establishment. For each sensor node i , all its neighbors can be categorized into two disjoint lists, which are the list of ordinary neighbors \mathcal{N}_i and the list of anchor neighbors \mathcal{M}_i , respectively. The total numbers of the ordinary neighbors and the anchor neighbors are denoted by $l_{\mathcal{N}_i}$ and $l_{\mathcal{M}_i}$ (lengths of the two lists). For convenience, refer to k th ordinary neighbor and k th anchor neighbor as $\mathcal{N}_i[k]$ and $\mathcal{M}_i[k]$ for sensor node i , with $k \in [1, l_{\mathcal{N}_i}]$, $k \in [1, l_{\mathcal{M}_i}]$. Denote the actual location of

sensor node i by r_i , and then the state X_i is established by the location vectors of both node i and all its neighbors

$$X_i = \left[r_i^T, r_{\mathcal{N}_i[1]}^T, \dots, r_{\mathcal{N}_i[L_{\mathcal{N}_i}]}^T \right]^T \quad (14)$$

whose dimension is $2 * (1 + L_{\mathcal{N}_i})$. Write the estimated location vector of the sensor node j which is estimated by sensor node i as $\hat{r}_{i \rightarrow j}$, and then the estimated state \hat{X}_i can be written as

$$\hat{X}_i = \left[\hat{r}_{i \rightarrow i}^T, \hat{r}_{i \rightarrow \mathcal{N}_i[1]}^T, \dots, \hat{r}_{i \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}^T \right]^T. \quad (15)$$

Denote the covariance of the state vector by P_i and the subblock matrix covariance corresponding to \hat{r}_j by $P_i \langle j \rangle$. It is clear that \hat{r}_j by $P_i \langle j \rangle$ is a main diagonal block of P_i

$$\begin{aligned} & \text{diag} \left(P_i \langle i \rangle, P_i \langle \mathcal{N}_i[1] \rangle, \dots, P_i \langle \mathcal{N}_i[L_{\mathcal{N}_i}] \rangle \right) \\ & \equiv \text{diag} (P_i). \end{aligned} \quad (16)$$

Because the sensor nodes being studied are static, the following differential equation is suitable for describing the change of the state

$$\dot{X}_i = \left[0^T, 0^T, \dots, 0^T \right]^T. \quad (17)$$

State Prediction. By (17), the prediction of the state and covariance at time $t = k$ can be given by

$$\begin{aligned} \hat{X}_i|_{k/k-1} &= \hat{X}_i|_{k-1}, \\ P_i|_{k/k-1} &= P_i|_{k-1} + Q_i. \end{aligned} \quad (18)$$

Local Filtering. At the time stamp $t = k$, the measurement vector is formed by all the neighboring distance measurements

$$\tilde{Z}_i|_k = \left[\tilde{d}_{i\mathcal{N}_i[1]}, \dots, \tilde{d}_{i\mathcal{N}_i[L_{\mathcal{N}_i}]}, \tilde{d}_{i\mathcal{M}_i[1]}, \dots, \tilde{d}_{i\mathcal{M}_i[L_{\mathcal{M}_i}]} \right]^T \quad (19)$$

whose dimension is $L_{\mathcal{N}_i} + L_{\mathcal{M}_i}$.

Similar to the Kalman filter, the local minimum variance estimate of the state immediately following the measurement can be obtained as

$$X_i|_{k-0.5} = X_i|_{k/k-1} + K_i|_k \left(\tilde{Z}_i|_k - \hat{Z}_i|_{k/k-1} \right). \quad (20)$$

The covariance matrix associated with the local minimum variance estimate can be obtained as

$$\begin{aligned} P_i|_{k-0.5} &= (I - K_i|_k H_i|_k) P_i|_{k/k-1} (I - K_i|_k H_i|_k)^T \\ &+ K_i|_k R_i K_i|_k^T, \end{aligned} \quad (21)$$

where the Kalman gain matrix is given by

$$K_i|_k = P_i|_{k/k-1} H_i|_k \left(H_i|_k P_i|_{k/k-1} H_i|_k^T + R_i \right)^{-1}. \quad (22)$$

Note that the relationship equation between the noise-free measurement and the state vector is

$$\begin{aligned} Z_i = h_i(X_i) &= \left[|r_i - r_{\mathcal{N}_i[1]}|, \dots, |r_i - r_{\mathcal{N}_i[L_{\mathcal{N}_i}]}|, \right. \\ &\left. |r_i - r_{\mathcal{M}_i[1]}|, \dots, |r_i - r_{\mathcal{M}_i[L_{\mathcal{M}_i}]}| \right]. \end{aligned} \quad (23)$$

Therefore, the measurement sensitivity matrix $H_i|_k$ in (21) and (22) can be determined as

$$\begin{aligned} H_i|_k &= \left. \frac{\partial h_i(X_i)}{\partial X_i} \right|_{X_i = \hat{X}_i|_{k/k-1}} \\ &= \begin{bmatrix} \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[1]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[1]}|} & - \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[1]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[1]}|} \\ \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[2]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[2]}|} & - \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[2]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[2]}|} \\ \vdots & \ddots \\ \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}|} & - \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}|} \\ \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{M}_i[1]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{M}_i[1]}|} & \\ \vdots & \\ \frac{\hat{r}_{i \rightarrow i}^T - \hat{r}_{i \rightarrow \mathcal{M}_i[L_{\mathcal{M}_i}]}^T}{|\hat{r}_{i \rightarrow i} - \hat{r}_{i \rightarrow \mathcal{M}_i[L_{\mathcal{M}_i}]}|} & \end{bmatrix}. \end{aligned} \quad (24)$$

Neighborhood Block Fusion. In this stage, each ordinary node fuses the block of the estimated state with its neighbors. Without loss of generality, the average fusion is considered. The average weight is determined by the number of the neighbors, following

$$\omega_i = \frac{1}{1 + L_{\mathcal{N}_i}}. \quad (25)$$

The ego's location is fused following

$$\begin{aligned} \hat{\mathbf{r}}_{i \rightarrow i}|_{k, \text{block}} \\ = P_i \langle i \rangle|_{k, \text{block}} \sum_{j \in \{i\} \cup \mathcal{N}_i} \left[\omega_j (P_j \langle j \rangle|_{k-0.5})^{-1} \hat{\mathbf{r}}_{j \rightarrow i}|_{k-0.5} \right] \end{aligned} \quad (26)$$

and the corresponding subblock covariance is fused following

$$(P_i \langle i \rangle|_{k, \text{block}})^{-1} = \sum_{j \in \{i\} \cup \mathcal{N}_i} \left[\omega_j (P_j \langle j \rangle|_{k-0.5})^{-1} \right]. \quad (27)$$

State and Covariance Reconstruction. In this stage, each ordinary node reconstructs the state estimate and the covariance by combining the fused blocks. The reconstructed state is given by

$$\begin{aligned} \hat{\mathbf{X}}_i|_k = \left[\hat{\mathbf{r}}_{i \rightarrow i}|_{k, \text{block}}^T, \hat{\mathbf{r}}_{\mathcal{N}_i[1] \rightarrow \mathcal{N}_i[1]}|_{k, \text{block}}^T, \dots, \right. \\ \left. \hat{\mathbf{r}}_{\mathcal{N}_i[L_{\mathcal{N}_i}] \rightarrow \mathcal{N}_i[L_{\mathcal{N}_i}]}|_{k, \text{block}}^T \right]^T \end{aligned} \quad (28)$$

and the main diagonal block of the reconstructed covariance matrix is given by

$$P_i \langle j \rangle|_k = P_j \langle j \rangle|_{k, \text{block}}. \quad (29)$$

From the definition of the state, it is easy to see that the nondiagonal block of the state covariance has little correlation with those of the neighbors. Therefore, the reconstructed covariance is approximately given by

$$\begin{aligned} P_i|_k = \text{diag} \left(P_i \langle i \rangle|_{k, \text{block}}, P_{\mathcal{N}_i[1]} \langle \mathcal{N}_i[1] \rangle|_{k, \text{block}}, \dots, \right. \\ \left. P_{\mathcal{N}_i[L_{\mathcal{N}_i}]} \langle \mathcal{N}_i[L_{\mathcal{N}_i}] \rangle|_{k, \text{block}} \right). \end{aligned} \quad (30)$$

It is worthwhile to note that since all the diagonal blocks of $P_i|_k$ are determined by the intersection covariance algorithm they are approximately optimized. Because all the diagonal blocks of $P_i|_k$ are independent, the overall reconstructed covariance $P_i|_k$ is also the near-minimum. Furthermore, the positive definiteness of the reconstructed covariance matrix would guarantee the consecutive iterations of the CNKF.

It is also worthwhile to note that the CNKF is sensitive to the initial guess of the state. The reason is twofold. Firstly, the distance measurement is not a linear projection of the location but a highly nonlinear and nonconvex one. So, the

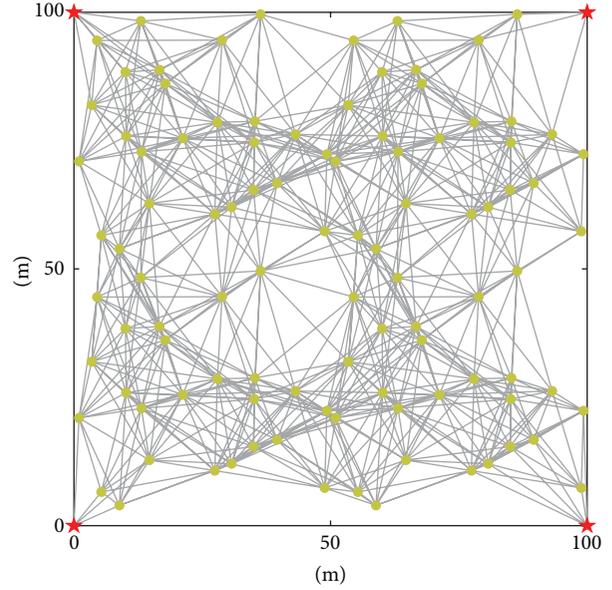


FIGURE 1: An instantiation of a wireless sensor network with a random topology. Red stars at the four corners represent the four anchor nodes, yellow circles inside the 100 m by 100 m square area represent the one hundred ordinary nodes, and grey lines indicate the communication relationships among nodes ($R_C = 25$ m).

solution might fall into the ambiguous or wrong optimum if the start point is not “close” enough to the true value. Secondly, The CNKF uses the first-order approximation to model the nonlinear measurement as shown in (24). So, the iterations may be hard to converge if the error of initial guess is large. Fortunately, some works (see, e.g., [26]) that focus on finding a good initial guess of the localization can be helpful. Therefore, using other algorithms first to find a good initial guess and then using CNKF to achieve much more accurate localization should be the best.

5. Simulation Result

In the experiments, we run CNKF algorithm on the WSNs with random topologies by numerical simulations. In the simulations, one hundred ordinary nodes are placed randomly inside a 100 m by 100 m square area, and another four anchor nodes are placed at the four corners of the square, respectively. As an instantiation, Figure 1 shows a distribution example of all the sensor nodes with the communication radius $R_C = 25$ m. In the graph, red stars represent the anchor nodes, yellow circles represent the ordinary nodes, and the grey lines among nodes indicate the communication relationships.

As a contrast algorithm, a well-known algorithm AFL by [15] is borrowed. The reason of comparing with the AFL is twofold. Firstly, the AFL is known to be able to achieve a relatively high accuracy localization compared with many other algorithms. Secondly, the AFL is also a fully simultaneous distributed algorithm just like the CNKF, while many other

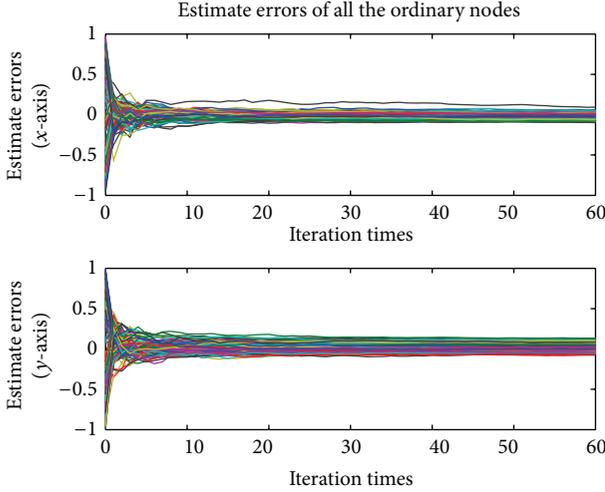


FIGURE 2: CNKF algorithm estimate errors of the one hundred ordinary nodes with different curves corresponding to different sensor nodes ($R_C = 25$ m).

algorithms are not fully distributed or simultaneous just as reviewed in Section 3.

The principle updating equation of the AFL comes from the mass-spring relaxation model, which can be given by

$$\hat{r}_i|_k = \hat{r}_i|_{k-1} + \sum_{j \in \mathcal{N}_i \cup \mathcal{M}_i} \alpha \cdot \tilde{n}_{ij}|_{k-1} (\hat{d}_{ij}|_{k-1} - \tilde{d}_{ij}|_k), \quad (31)$$

$$\forall i \in \mathcal{O},$$

where $\hat{d}_{ij} = |\hat{r}_i - \hat{r}_j|$ denotes the distance estimate between nodes i and j , $\tilde{n}_{ij} = -(\hat{r}_i - \hat{r}_j)/\hat{d}_{ij}$ denotes the unit vector from node i to node j , and α is a convergence coefficient (usually small in order to guarantee the convergence) influencing the converging speed and accuracy.

In the simulations, rough initial location estimates are given to all the ordinary sensors, whose errors are about 1 meter on each axis. Accordingly, the covariance matrices $\{P_i\}$ are all initialized to the identity matrix. The distance measurements are corrupted by Gaussian noises as in (12), in which the noise ratio $Y = 3$. The processing noise matrices $\{Q_i\}$ are set to diagonal matrices, whose diagonal elements are positive and very small, because the nodes in the network are static. The measurement noise matrices $\{R_i\}$ are also constructed to the diagonal matrices, whose diagonal elements are $\{\hat{\sigma}_{\varepsilon_{ij}}^2 \mid j \in \mathcal{N}_i \cup \mathcal{M}_i\}$ given by

$$\hat{\sigma}_{\varepsilon_{ij}} = \hat{d}_{ij} * \frac{Y}{100}. \quad (32)$$

Figure 2 shows the overall performance of the CNKF, where the curves represent the estimate errors of all the ordinary nodes in both the x - and y -axes. It can be seen that the estimate errors decrease gradually with the increase of the iteration times, implying that CNKF is effective in reducing the errors and refining the location estimates. In order to show more detailed results, two metrics are borrowed as

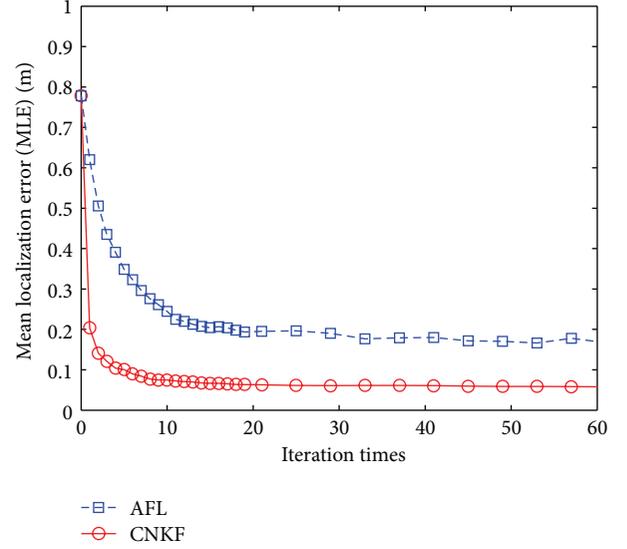


FIGURE 3: The mean localization errors for the CNKF and AFL algorithms ($R_C = 25$ m).

follows to evaluate the accuracy performance of localization algorithms.

Performance Metric 1 (mean localization error). The mean localization error (MLE) is used to indicate the individual localization error of all the ordinary nodes, given by

$$\text{MLE} = \frac{1}{n} \sum_{i=1}^n |\hat{r}_{i \rightarrow i} - r_i|. \quad (33)$$

Performance Metric 2 (global energy ratio). The global energy ratio (GER) is used to indicate the structural error of the localization of the network, given by

$$\text{GER} = \sqrt{\frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \left(\frac{|\hat{r}_{i \rightarrow i} - \hat{r}_{j \rightarrow j}| - d_{ij}}{d_{ij}} \right)^2}. \quad (34)$$

First of all, a common case is simulated to help see the improvement of the CNKF, and simulation conditions are set to be the same for both CNKF and AFL with $R_C = 25$ m and $\alpha = 0.03$. Figures 3 and 4 show the comparison results of the estimate errors of the two algorithms (during the first 60 iterations). The metrics of MLE and GER both decrease more rapidly and more sharply by using the CNKF, implying that the CNKF can quickly achieve the convergence with high accuracy in this case.

Furthermore, in order to see the steady-state performance of the CNKF, plenty more experiments under various cases are also simulated. We record the data after 2000 iterations as the steady-state results, and then we calculate the metrics of MLE and GER by averaging the data of the repeated experiments. Table 1 shows the corresponding comparison results. It can be seen from Table 1 that both the metrics of the MLE and GER are enhanced by at least 57%, which

TABLE 1: Comparison results of accuracy for the CNKF and AFL algorithms.

Conditions		MLE (m)			GER (%)		
R_C (m)	A_C	AFL	CNKF	Enhanced	AFL	CNKF	Enhanced
15	≈ 7	0.040	0.020	50%	0.15	0.05	67%
20	≈ 10	0.035	0.015	57%	0.15	0.05	67%
25	≈ 15	0.035	0.015	57%	0.15	0.05	67%
30	≈ 20	0.040	0.015	63%	0.20	0.05	75%
35	≈ 27	0.045	0.015	67%	0.25	0.05	80%
40	≈ 35	0.050	0.015	70%	0.25	0.05	80%

R_C : communication radius.

A_C : average connectivity.

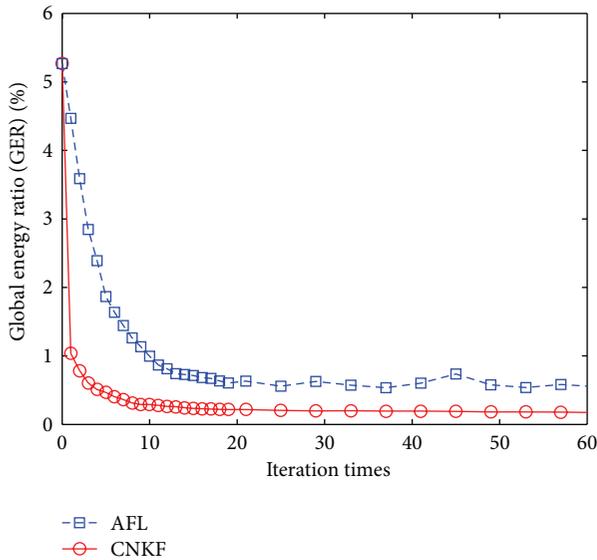


FIGURE 4: The global energy ratios for the CNKF and AFL algorithms ($R_C = 25$ m).

means that the CNKF can achieve a much smaller estimate error in comparison with the AFL or, in other words, the localization accuracy of the CNKF is significantly high. The reason behind the result mainly lies in the full use of the intranodes information, which treats the estimate as a distribution instead of a simple point following the data-fusion perspective, so that it allows the correlated estimates to make a much better refinement after fusion. It is worthwhile to note that the performance of the CNKF is more insensitive to the average connectivity of the network, which is of benefit in the applications of flexible and large-scale networks.

It is also necessary to point out that CNKF would probably diverge and fail to achieve the localization when the initial guess of the locations has a large deviation. So, in the experiments, we have assumed that a good guess (whose error is no more than 1 meter) is already available. As previously noted in Section 4, some initial algorithms or practical means could be used to provide such an initial guess.

6. Conclusion

This paper deals with the localization problems of WSNs and proposes a fully distributed localization algorithm named

CNKF. The CNKF is based on the concept of data fusion, allowing all the intranodes information to take part in the estimating process, and it has the capacity of estimating sensor locations in a fast speed and a high accuracy. The key points of the algorithm are fourfold. The first point is the constructing way to establish the appropriate form of the states, so that neighboring nodes can work under a collaborative mechanism. The second point is the use of the static nature of the network to predict the states. The third point is the filtering and fusing way to refine the estimated states both locally and cooperatively. The fourth point is the reconstructing way to update the states and the corresponding covariance matrices, so that the optima and consecution of the iterative estimation can be guaranteed. Finally, in order to validate the algorithm, a series of experiments including the comparison are performed by the simulations. The results show that the proposed algorithm is effective and suitable for the wireless sensor networks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Thanks are due to the support by the National Natural Science Foundation of China (Grant no. 61403392).

References

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [2] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Courier Corporation, 2007.
- [3] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982.
- [4] S. J. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection (ci)," in *Handbook of Data Fusion*, D. Hall and J. Llinas, Eds., CRC Press, Boca Raton, Fla, USA, 2001.
- [5] Mansoor-ul-Haque, F. A. Khan, and M. Iftikhar, "Optimized energy-efficient iterative distributed localization for wireless sensor networks," in *Proceedings of the IEEE International*

- Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 1407–1412, IEEE, October 2013.
- [6] 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 Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.
 - [7] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
 - [8] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacomo, "A review of localization algorithms for distributed wireless sensor networks in manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 7, pp. 698–716, 2009.
 - [9] A. Pal, "Localization algorithms in wireless sensor networks: current approaches and future challenges," *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 45–73, 2010.
 - [10] A. Panwar and S. A. Kumar, "Localization schemes in wireless sensor networks," in *Proceedings of the 2nd International Conference on Advanced Computing and Communication Technologies (ACCT '12)*, pp. 443–449, IEEE, Rohtak, India, January 2012.
 - [11] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, no. 4, pp. 2419–2436, 2013.
 - [12] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 5, pp. 2926–2931, IEEE, San Antonio, Tex, USA, November 2001.
 - [13] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 516–526, 2010.
 - [14] Z. He, Y. Ma, and R. Tafazolli, "Cooperative localization in a distributed base station scenario," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC Spring '11)*, pp. 1–5, IEEE, Budapest, Hungary, May 2011.
 - [15] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Poster abstract: anchor-free distributed localization in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 340–341, Los Angeles, Calif, USA, November 2003.
 - [16] M. Xia, P. Sun, X. Wang, Y. Jin, and Q. Chen, "Distributed beacon drifting detection for localization in unstable environments," *Mathematical Problems in Engineering*, vol. 2013, Article ID 865983, 8 pages, 2013.
 - [17] L. Wen, L. Cui, S. Chai, and B. Zhang, "Neighbor constraint assisted distributed localization for wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2014, Article ID 143938, 11 pages, 2014.
 - [18] U. A. Khan, S. Kar, and J. M. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.
 - [19] U. A. Khan, S. Kar, and J. M. F. Moura, "DILAND: an algorithm for distributed sensor localization with noisy distance measurements," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1940–1947, 2010.
 - [20] S. Zhu and Z. Ding, "Distributed cooperative localization of wireless sensor networks with convex hull constraint," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2150–2161, 2011.
 - [21] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 45–57, ACM, October 2004.
 - [22] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Proceedings of the 6th IEEE International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 51–60, Cambridge, Mass, USA, April 2007.
 - [23] M. Z. Win, A. Conti, S. Mazuelas et al., "Network localization and navigation via cooperation," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 56–62, 2011.
 - [24] R. Olfati-Saber, "Kalman-consensus filter: optimality, stability, and performance," in *Proceedings of the 48th IEEE Conference on Decision and Control Held Jointly with the 28th Chinese Control Conference (CDC/CCC '09)*, pp. 7036–7042, IEEE, Shanghai, China, December 2009.
 - [25] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.
 - [26] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 50–61, ACM, Baltimore, Md, USA, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

