

Research Article

Distributed Filter with Consensus Strategies for Sensor Networks

Xie Li, Huang Caimou, and Hu Haoji

Faculty of Information, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Hu Haoji; haoji.hu@zju.edu.cn

Received 12 August 2013; Accepted 28 August 2013

Academic Editor: Baocang Ding

Copyright © 2013 Xie Li 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.

Consensus algorithm for networked dynamic systems is an important research problem for data fusion in sensor networks. In this paper, the distributed filter with consensus strategies known as Kalman consensus filter and information consensus filter is investigated for state estimation of distributed sensor networks. Firstly, an in-depth comparison analysis between Kalman consensus filter and information consensus filter is given, and the result shows that the information consensus filter performs better than the Kalman consensus filter. Secondly, a novel optimization process to update the consensus weights is proposed based on the information consensus filter. Finally, some numerical simulations are given, and the experiment results show that the proposed method achieves better performance than the existing consensus filter strategies.

1. Introduction

In recent years, there has been a surge of interests in the area of distributed sensor networks. The advantages of distributed sensor networks lie in their low processing power, cheap memory, scalable sensing features, and fault tolerance capabilities.

One of the most basic problems for distributed sensor networks is to develop distributed algorithms [1] for the state estimation of a process of interest. When a process is observed by a group of sensors organized in a network, the goal of each sensor node is to obtain the accurate state estimation for the process. Kalman filtering has been proved to be an effective algorithm for state estimation of dynamic processes [2, 3]. Because of this, most papers focusing on distributed estimation propose different mechanisms by combining the Kalman filter with a consensus filter in order to ensure that the estimates asymptotically converge to the same value, schemes which will be henceforth called consensus based distributed filtering algorithms. Based on the idea mentioned above, a scheme for *distributed Kalman filtering* (DKF) was proposed in [4] based on reaching an average-consensus [5, 6], and in [7] Olfati-Saber proposed a scalable and distributed Kalman filtering algorithm based on reaching a dynamic average consensus [8]. Olfati-Saber's algorithm [7] has been

further developed by other researchers [9] with similar algorithms. However, methods based on such kind of algorithms produce relatively weak performances. According to [10], the performance is compared with the collective estimation error of n noncooperative local Kalman filters, which is a trivial base performance level for distributed estimation in sensor networks. To solve this, Olfati-Saber developed the *Kalman consensus filter* (KCF) in [11], where a consensus filter runs directly on the estimator state space variables. In addition, a formal derivation followed by optimality and stability analysis of KCF in discrete-time has been elaborated in [10]. However, in distributed implementations, there is a correlation between local estimates [12] in KCF. In general distributed networks, it is not possible to exactly determine this correlation [13] and it results in nonoptimal local estimates. Other techniques to accomplish distributed estimation for dynamic systems that rely on the inverse covariance filter or information filter have been around for many years [14, 15]. An *information consensus filter* (ICF) is presented in [16] that applies consensus filters to an information filter. This method does not exactly solve the problem of correlation between local estimates but it gives insight into the statistical effects of the correlation and is working much well in distributed sensor networks. Based on the ICF, we focus on designing the consensus weights to improve its performance.

In this paper, we firstly describe the existing distributed filter with consensus strategies. Then we make an in-depth comparison between the KCF and the ICF. Based on the ICF, we propose the consensus weights optimization for better performance of the system and refer this method as *weights optimized information consensus filter* (WO-ICF). We show experimentally that the proposed method achieves the best performance and it is closest to the optimal centralized performance.

The structure of this paper is organized as follows. In Section 2, the background knowledge on Kalman filter and the centralized information filter are provided. In Section 3, the consensus strategies are discussed. In Section 4, Kalman consensus filter is presented. In Section 5 we discuss the information consensus filter, and an ICF based weights optimization method is proposed. Simulation results and performance comparisons between the KCF and ICF algorithm are provided in Section 6. In Section 7, we make a brief summary.

2. Kalman Filter: Information Form

2.1. Kalman Filter. Consider a dynamic process with the linear time-varying model as follows:

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)w(k); \\ x(0) &\in N(\bar{x}(0), P_0), \end{aligned} \quad (1)$$

where $x(k) \in R^n$ and $w(k) \in R^m$ are the state and input noise of the process at time $k \in \{0, 1, 2, \dots\}$, respectively; $x(0)$ is the initial state with a Gaussian distribution; $A(k)$ is the model matrix, $B(k)$ is the state noise matrix. We are interested in tracking the state of this target by the use of a sensor network with n sensors and the communication topology $G = (V; E)$.

The observations at sensor i and time k are

$$z_i(k) = H_i(k)x(k) + v_i(k), \quad (2)$$

where $z_i(k) \in R^{p_i}$ with $\sum_{i=1}^n p_i = p$, $H_i(k) \in R^{p_i \times n}$ is the local observation matrix for sensor i , and $v_i(k)$ is the local observation noise. We refer to $z_i(k)$ as sensor data. Assume that $w(k)$ and $v_i(k)$ are zero mean white Gaussian noise with the following statistics:

$$\begin{aligned} E[w(k)w(l)^T] &= Q(k)\delta_{kl}, \\ E[v_i(k)v_i(l)^T] &= R_i(k)\delta_{kl}, \end{aligned} \quad (3)$$

where $\delta_{rs} = 1$ if $r = s$ and $\delta_{rs} = 0$, otherwise. We stack the observations at all n sensors in the sensor network to get the global observation model as follows.

Let the global observation vector $\mathbf{z}(k) \in R^p$, the global observation matrix $H(k) \in R^{p \times n}$, and the global observation noise vector $\nu(k) \in R^n$ be

$$\begin{aligned} \mathbf{z}(k) &= [z_1(k), z_2(k), \dots, z_n(k)]^T, \\ H(k) &= [H_1(k), H_2(k), \dots, H_n(k)]^T, \\ \nu(k) &= [v_1(k), v_2(k), \dots, v_n(k)]^T. \end{aligned} \quad (4)$$

Then the global observation model is given by

$$\mathbf{z}(k) = H(k)x(k) + \nu(k). \quad (5)$$

Since observation noises of different sensors are mutually independent, we can combine $R_i(k)$ into one global observation noise covariance matrix $R(k)$ as

$$R(k) = \text{block diag}[R_1(k), \dots, R_n(k)]. \quad (6)$$

Given the collective information $Z(k) = \{\mathbf{z}(0), \mathbf{z}(1), \dots, \mathbf{z}(k)\}$, the estimation of the state of the process can be expressed as

$$\begin{aligned} \hat{x}(k) &:= \hat{x}(k | Z(k)) = E[x(k) | Z(k)], \\ \bar{x}(k) &:= \hat{x}(k | Z(k-1)) = E[x(k) | Z(k-1)]. \end{aligned} \quad (7)$$

We refer to $\hat{x}(k)$ and $\bar{x}(k)$ as estimate and prior estimate (or prediction) of the state $x(k)$, respectively. Then, the error covariance matrices associated with the estimates $\hat{x}(k)$ and $\bar{x}(k)$ are given by

$$\begin{aligned} M(k) &:= E[(\hat{x}(k) - x(k))(\hat{x}(k) - x(k))^T] = E[\eta(k)\eta(k)^T], \\ P(k) &:= E[(\bar{x}(k) - x(k))(\bar{x}(k) - x(k))^T] = E[\bar{\eta}(k)\bar{\eta}(k)^T], \end{aligned} \quad (8)$$

where $\eta(k) = \hat{x}(k) - x(k)$ and $\bar{\eta}(k) = \bar{x}(k) - x(k)$ denote the estimate errors and $P(0) = P_0$. Then, the Kalman filter is a linear estimator in the form

$$\hat{x}(k) = \bar{x}(k) + K(k)(\mathbf{z}(k) - H(k)\bar{x}(k)) \quad (9)$$

with the Kalman gain $K(k)$.

Remark 1. Throughout this paper, due to the importance of the node indices, we adopt a notation that is free of the time-index k and call it an index-free notation to represent all estimators [10]. The index-free form of the above estimator can be written as

$$\hat{x} = \bar{x} + K(\mathbf{z} - H\bar{x}). \quad (10)$$

We also use the update operation $\{\cdot\}^+$ defined in [10] to rewrite the sensing model of node i of the sensor network as

$$x^+ = Ax + Bw, \quad z_i = H_i x + v_i. \quad (11)$$

Then, we get the index-free recursive equations of a *centralized Kalman filter* (CKF) for system (11):

$$\begin{aligned} \hat{x} &= \bar{x} + K(\mathbf{z} - H\bar{x}), \\ K &= PH^T(R + HPH^T)^{-1}, \\ M &= P - PH^T(R + HPH^T)^{-1}HP, \\ P^+ &= AMA^T + BQB^T, \\ \bar{x}^+ &= A\hat{x}. \end{aligned} \quad (12)$$

2.2. *Information Form: Centralized Information Filter.* Using the matrix inversion lemma

$$\begin{aligned} (I + ABC^{-1}B^T)^{-1}A &= (A^{-1} + BC^{-1}B^T)^{-1} \\ &= A - AB(B^T AB + C)^{-1}B^T A. \end{aligned} \quad (13)$$

By use of the identity $B^T ABC^{-1} = [(B^T AB + C)C^{-1} - I]$, we have

$$\begin{aligned} (I + ABC^{-1}B^T)^{-1}ABC^{-1} &= (A^{-1} + BC^{-1}B^T)^{-1}BC^{-1} \\ &= AB(B^T AB + C)^{-1}. \end{aligned} \quad (14)$$

From (14), we have

$$K = PH^T(R + HPH^T)^{-1} = (P^{-1} + H^T R^{-1}H)^{-1}H^T R^{-1}. \quad (15)$$

Using the matrix inversion lemma

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}, \quad (16)$$

we have

$$(P^{-1} + H^T R^{-1}H)^{-1} = P - PH^T(R + HPH^T)^{-1}HP = M. \quad (17)$$

Then we can rewrite (15) as

$$K = MH^T R^{-1}. \quad (18)$$

Based on the above derivation, the recursive equations of the Kalman filter can be rewritten as

$$\begin{aligned} \hat{x} &= \bar{x} + K(\mathbf{z} - H\bar{x}), \\ K &= MH^T R^{-1}, \\ M &= (P^{-1} + H^T R^{-1}H)^{-1}, \\ P^+ &= AMA^T + BQB^T, \\ \bar{x}^+ &= A\hat{x}. \end{aligned} \quad (19)$$

From (19), the estimate \hat{x} can be expressed as

$$\begin{aligned} \hat{x} &= \bar{x} + K(\mathbf{z} - H\bar{x}) = \bar{x} + MH^T R^{-1}(\mathbf{z} - H\bar{x}) \\ &= \bar{x} + M(H^T R^{-1}\mathbf{z} - H^T R^{-1}H\bar{x}). \end{aligned} \quad (20)$$

Now we define the n -dimensional global observation variables as

$$\mathbf{y} = H^T R^{-1}\mathbf{z}, \quad S = H^T R^{-1}H \quad (21)$$

and the n -dimensional local observation variables at sensor i as

$$y_i = H_i^T R_i^{-1}z_i, \quad S_i = H_i^T R_i^{-1}H_i. \quad (22)$$

When the observations are distributed among the sensors, the KF can be implemented by collecting all the sensor observations at a central location, or with observation fusion by realizing that the global observation variables in (21), it can be written as

$$\mathbf{y} = H^T R^{-1}\mathbf{z} = H_1^T R_1^{-1}\mathbf{z}_1 + \cdots + H_N^T R_N^{-1}\mathbf{z}_N = \sum_{i=1}^N y_i. \quad (23)$$

Similarly,

$$S = H^T R^{-1}H = H_1^T R_1^{-1}H_1 + \cdots + H_N^T R_N^{-1}H_N = \sum_{i=1}^N S_i. \quad (24)$$

Recall (20) where the estimate \hat{x} could be written as

$$\hat{x} = \bar{x} + M(\mathbf{y} - S\bar{x}). \quad (25)$$

Multiplication on the left by M^{-1} yields a variation of (25) as

$$M^{-1}\hat{x} = M^{-1}\bar{x} + \mathbf{y} - S\bar{x} = (P^{-1} + S)\bar{x} + \mathbf{y} - S\bar{x} = P^{-1}\bar{x} + \mathbf{y}. \quad (26)$$

Let the inverse of M and P be the information matrices, \hat{I} and \bar{I} . Let \hat{i} and \bar{i} be the information vectors. We have the following relations:

$$\begin{aligned} \hat{I} &= M^{-1}, & \bar{I} &= P^{-1}, \\ \hat{i} &= M^{-1}\hat{x}, & \bar{i} &= P^{-1}\bar{x}. \end{aligned} \quad (27)$$

Then (26) can be rewritten as

$$\hat{i} = \bar{i} + \mathbf{y}, \quad (28)$$

and the update of predicted estimate \bar{x} can be expressed as

$$\bar{x}^+ = A\hat{x} = A(M\hat{I})\hat{x} = AM(\hat{I}\hat{x}) = A\hat{I}^{-1}\hat{i}, \quad (29)$$

and we have

$$\bar{i}^+ = (P^{-1}\bar{x})^+ = \bar{I}^+ A\hat{I}^{-1}\hat{i}. \quad (30)$$

Now we get the following simpler form of the filter in (19) and call it centralized information filter (CIF):

$$\begin{aligned} \hat{I} &= \bar{I} + S, & \hat{i} &= \bar{i} + \mathbf{y}, \\ \bar{I}^+ &= (A\hat{I}^{-1}A^T + BQB^T)^{-1}, & \bar{i}^+ &= \bar{I}^+ A\hat{I}^{-1}\hat{i}, \end{aligned} \quad (31)$$

where (31) is the filter step and prediction step (or update step) of the CIF, respectively.

3. Consensus Strategy

Consensus strategy defines a set of rules for a team of agents to agree on specific consensus states. With these rules each agent exchanges information with its neighboring agents and finally reaches an agreement (or consensus) concerning the consensus state over time [17, 18]. Furthermore, average

consensus occurs when the final consensus state is the average of the initial values.

Consider a team of n agents to agree on specific consensus states, and at any discrete-time instant τ , the communication topology between n agents can be described by the graph $G[\tau] = (V, E[\tau])$, the graph G is undirected, $V = \{1, 2, \dots, n\}$ is the vertex set, and $E[\tau] \subseteq V \times V$ is the edge set. In the consensus algorithm, each agent in the network maintains a local copy of the consensus state $\zeta_i \in \mathbb{R}^n$ and updates this value using its neighbors' consensus states according to the rule:

$$\zeta_i[\tau + 1] = \zeta_i[\tau] + \sum_{j=1}^n \beta_{ij}[\tau] (\zeta_j[\tau] - \zeta_i[\tau]), \quad (32)$$

where τ indicates the consensus filter iteration step. To choose the weights $\beta_{ij}[\tau]$, we can use the Maximum-degree weights or the Metropolis weights [19]. Here we use the latter which preserves the averaging in consensus filters and can be computed by

$$\beta_{ij}[\tau] = \begin{cases} \left(1 + \max\{d_i[\tau], d_j[\tau]\}\right)^{-1} & \text{if } (i, j) \in E[\tau], \\ 1 - \sum_{(i,l) \in E[\tau]} \beta_{il}[\tau] & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad (33)$$

where $d_i[\tau]$ is the degree of agent i in the graph $G[\tau]$. Arrange the local consensus states into the vector $\zeta[\tau] = [\zeta_1^T[\tau], \dots, \zeta_n^T[\tau]]^T$, and define the matrix $(B[\tau])_{ij} = \beta_{ij}[\tau]$ for $i \neq j$; otherwise $(B[\tau])_{ii} = 1 - \sum_{(i,l) \in E[\tau]} \beta_{il}[\tau]$, and we can rewrite the update in (32) as

$$\zeta[\tau + 1] = (B[\tau] \otimes I) \zeta[\tau], \quad (34)$$

where I is the appropriate size identity matrix and \otimes denotes the matrix Kronecker product.

The ij th element of $B[\tau]$ in (34) satisfies the following four conditions: (1) $(B[\tau])_{ij} \geq 0$, (2) $\sum_i (B[\tau])_{ij} = 1$, (3) $\sum_j (B[\tau])_{ij} = 1$, (4) and each nonzero entry is both uniformly upper and lower bounded. Based on these conditions, we have the following results [20] for average consensus.

Lemma 2. *Under switching interaction topologies, if there exists a finite $T \geq 0$ that for every interval $[\tau, \tau+T]$ the union of the interaction graph across interval is strongly connected, then consensus protocol (34) achieves average consensus asymptotically; that is, $\zeta_i[\tau] \rightarrow (1/n) \sum_{i=1}^n \zeta_i[0]$ as $\tau \rightarrow \infty$.*

Remark 3. In order to calculate the metropolis weights in (33), we assume undirected communication throughout this paper. Therefore, if the graph $G[\tau]$ is connected, the matrix $B[\tau]$ is a doubly stochastic matrix, and the four conditions on $B[\tau]$ are satisfied. This implies that average consensus is achieved asymptotically as long as every graph is connected [21].

4. Distributed Kalman Filter: Consensus on Estimate

In this section, we discuss an alternative approach to distribute the Kalman filtering that relies on communicating state estimates between neighboring nodes and refer to it as *Kalman consensus filter* (KCF). Before presenting the KCF algorithm, we first need to discuss a more primitive DKF algorithm called local Kalman filter (LKF) which forms the basis of the KCF.

4.1. Local Kalman Filter. In local Kalman filtering, let $N_i = \{j : (i, j) \in E\}$ be the set of neighbors of node i on graph G . Each node i of the sensor network communicates its measurement z_i , covariance information R_i , and observation matrix H_i with its neighbors N_i . For node i , we assume that the information flow from nonneighboring nodes to node i is prohibited if there is no nodes except for its neighbors N_i exist. Therefore, node i can use a central Kalman filter that only utilizes the observation vectors and observation matrices of the nodes in $J_i = N_i \cup \{i\}$ [11]. This leads to the following primitive DKF algorithm with no consensus on state estimation.

LKF Iterations. Assume that node i only receives information from its neighbors. Then, we have the iterations of node i in local Kalman filtering as

$$\begin{aligned} y^i &= \sum_{j \in J_i} H_j^T R_j^{-1} z_j = \sum_{j \in J_i} y_j, \\ S^i &= \sum_{j \in J_i} H_j^T R_j^{-1} H_j = \sum_{j \in J_i} S_j, \\ \hat{x}_i &= \bar{x}_i + M_i (y^i - S^i \bar{x}), \end{aligned} \quad (35)$$

$$M_i = (P_i^{-1} + S^i)^{-1},$$

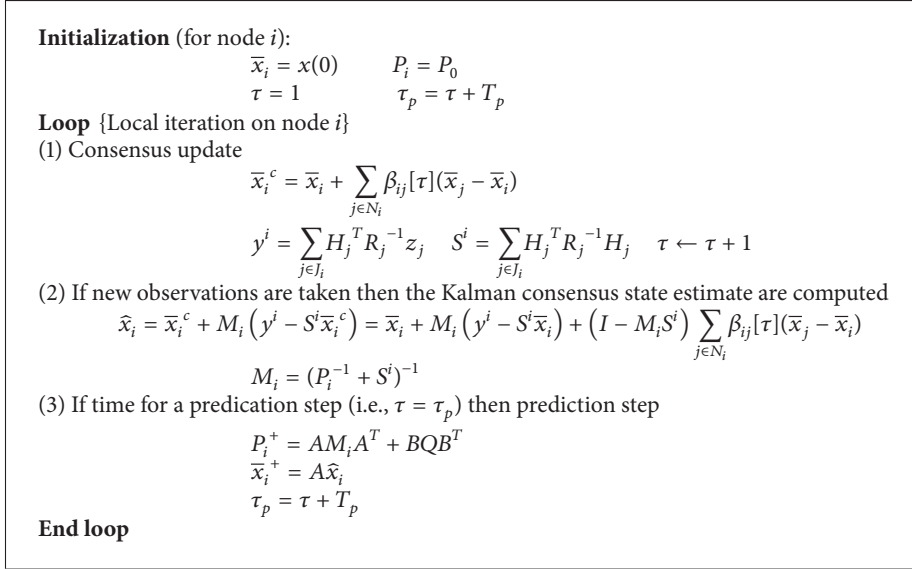
$$P_i^+ = A M_i A^T + B Q B^T,$$

$$\bar{x}_i^+ = A \hat{x}_i,$$

where y^i and S^i are local aggregate information vector and matrix, respectively and node i locally computes both y^i and S^i .

4.2. Kalman Consensus Filter. We now present the Kalman consensus filter (KCF). The KCF uses consensus strategy (32) on the state estimate in a distributed Kalman filter, where each node maintains a local Kalman filter. Corresponding to (32), let $\zeta_i[\tau] = \bar{x}_i$ be the prior estimate at time τ and $\zeta_i[\tau+1] = \bar{x}_i^c$ be fused prior estimate; each node fuses the prior estimates from its neighbors according to the rule:

$$\bar{x}_i^c = \bar{x}_i + \sum_{j \in N_i} \beta_{ij}[\tau] (\bar{x}_j - \bar{x}_i). \quad (36)$$



ALGORITHM 1: Kalman consensus filter.

Using the fused prior estimate \bar{x}_i^c , the filter estimate at node i could be implemented by

$$\begin{aligned} \hat{x}_i &= \bar{x}_i^c + M_i (y^i - S^i \bar{x}_i^c) \\ &= \bar{x}_i + M_i (y^i - S^i \bar{x}_i) + (I - M_i S^i) \sum_{j \in N_i} \beta_{ij}[\tau](\bar{x}_j - \bar{x}_i). \end{aligned} \quad (37)$$

The local KCF is summarized in Algorithm 1, where τ is the time index for the consensus strategy and $T_p \in \mathbb{Z}^+$ is the time interval between prediction updates. One-time step $k - 1 \rightarrow k$ is equivalent to T_p time steps of the consensus time index $\tau \rightarrow \tau + 1$; that is, for each node, the information exchanges between neighboring nodes occurred faster than the prediction update step. The three steps in KCF prediction, local filter estimate, and consensus update are not necessarily sequential.

The last term in (37) is the correction of filter estimate \hat{x}_i compared to the standard Kalman estimator. Intuitively, adding the consensus term in (37) will force local estimators to reach a consensus regarding state estimates. The structure of node i in the KCF algorithm is shown in Figure 1.

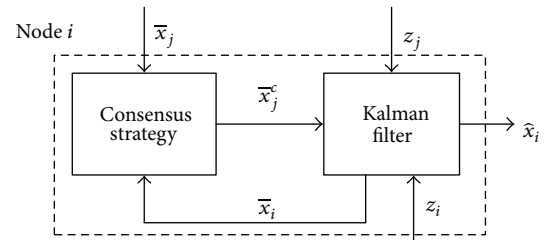
In [11], the author proposed the following Kalman consensus estimator which is in the form of

$$\hat{x}_i = \bar{x}_i + M_i (y^i - S^i \bar{x}_i) + C_i \sum_{j \in N_i} (\bar{x}_j - \bar{x}_i), \quad (38)$$

where C_i is named consensus gain of node i . The choice of the consensus gain C_i is free. A poor choice of C_i leads to either the lack of consensus on estimates (e.g., setting $C_i = 0$, for all i) or the lack of stability of the error dynamics of the filter. One possible choice is to let

$$C_i = \gamma P_i = \varepsilon \frac{P_i}{1 + \|P_i\|_F}, \quad (39)$$

where $\varepsilon > 0$ is a relative small constant and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The derivation of the optimal

FIGURE 1: The algorithm structure of node i in the KCF.

Kalman consensus filter can be found in [10], where we can also find that the computational complexity of updating the error covariance $P_i^+ = A M_i A^T + B Q B^T$ of the optimal Kalman consensus filter is not scalable in n . To obtain a suboptimal approximation of the KCF which is distributed and scalable in n , we make an assumption that the consensus gains $C_i = O(\varepsilon)$ are of the order of ε . Then we get the stable suboptimal KCF summarized in Algorithm 2.

5. Distributed Filter: Consensus on Information Matrix

The KCF discussed previously applies consensus strategy on the prior estimate to the Kalman filter and improves the state estimate of KF. However, the error covariance matrix M_i is not improved because each node in KCF only fuses the prior estimates from its neighbors but neglects the helpful information about the error covariance matrix. In the next section, we adopt an information matrix weighted consensus strategy to improve the consensus based distributed Kalman filter algorithm in the estimation fusion of sensor networks. We refer to this method as information consensus filter (ICF) [16]. Before presenting the ICF algorithm, we need to discuss a more primitive DKF algorithm called *local information filter* (LIF) which forms the basis of the ICF.

Initialization: $\bar{x}_i = x(0), P_i = P_0,$ and message $m_j = \{y_j, S_j, \bar{x}_j\}$

While new data exists **do**

(1) Compute local observation vector and matrix of node i :

$$y_i = H_i^T R_i^{-1} z_i \quad S_i = H_i^T R_i^{-1} H_i$$

(2) Broadcast message $m_i = \{y_i, S_i, \bar{x}_i\}$ to neighbors.

(3) Receive messages from all neighbors.

(4) Compute the local aggregate information vector and matrix:

$$y^i = \sum_{j \in J_i} y_j \quad S^i = \sum_{j \in J_i} S_j$$

(5) Compute the Kalman consensus state estimate

$$\hat{x}_i = \bar{x}_i + M_i(y^i - S^i \bar{x}_i) + \gamma P_i \sum_{j \in N_i} (\bar{x}_j - \bar{x}_i)$$

$$M_i = (P_i^{-1} + S^i)^{-1}$$

$$\gamma = \frac{\epsilon}{1 + \|P_i\|}, \quad \|X\| = \text{tr}(X^T X)^{1/2}$$

(6) Update the state of the Kalman consensus filter:

$$P_i^+ = A M_i A^T + B Q B^T$$

$$\bar{x}_i^+ = A \hat{x}_i$$

End While

ALGORITHM 2: Suboptimal Kalman consensus filter: DKF Algorithm with an estimator that has a rigorously derived consensus term (message passing during one time cycle for node).

5.1. Local Information Filter. To distribute the estimation of the global state vector, x in CIF, we implement local information filter (LIF) at each sensor i , which is based on the sensor model (11) and can be derived from local Kalman filter (LKF) in (35) after we use its information form. Each LIF computes local objects (matrices and vectors) which are then fused (if required) by exchanging information among the neighbors. In LIF, there is no centralized knowledge of the estimation of the global state that exists in CIF; however, it can be obtain by fusing the local state vector.

Let the inverses of M_i and P_i be the local information matrices, \hat{I}_i and \bar{I}_i . Let \hat{i}_i and \bar{i}_i be the local information vector. We have the following relations:

$$\begin{aligned} \hat{I}_i &= M_i^{-1}, & \bar{I}_i &= P_i^{-1}, \\ \hat{i}_i &= M_i^{-1} \hat{x}_i, & \bar{i}_i &= P_i^{-1} \bar{x}_i. \end{aligned} \quad (40)$$

Then we get the following simpler form of the filter in (35).

LIF Iterations. The local information filtering iterations for node i are in the form

$$\begin{aligned} \hat{I}_i &= \bar{I}_i + S^i, & \hat{i}_i &= \bar{i}_i + y^i, \\ \hat{I}_i^+ &= (A \hat{I}_i^{-1} A^T + B Q B^T)^{-1}, & \hat{i}_i^+ &= \bar{I}_i^+ A \hat{I}_i^{-1} \hat{i}_i. \end{aligned} \quad (41)$$

5.2. Information Consensus Filter. We now present the information consensus filter (ICF). The ICF uses consensus strategy (32) on both the information state and the information matrix in a distributed Kalman filter, where each node maintains a local information filter. Recalling (32), let \bar{i}_i^c be the fused local information vector and \bar{I}_i^c the fused local

information matrix; each node fuses the local information from its neighbors according to the rule:

$$\begin{aligned} \bar{i}_i^c &= \bar{i}_i + \sum_{j \in N_i} \beta_{ij}[\tau] (\bar{i}_j - \bar{i}_i), \\ \bar{I}_i^c &= \bar{I}_i + \sum_{j \in N_i} \beta_{ij}[\tau] (\bar{I}_j - \bar{I}_i). \end{aligned} \quad (42)$$

Using the fused local information vector and matrix, \bar{i}_i^c and \bar{I}_i^c , the local ICF is summarized in Algorithm 3.

5.3. Local Information Filter. Now we make a comparison between ICF and KCF based on the state estimate \hat{x}_i and the error covariance matrix M_i . Let $\beta_{ii}[\tau] = 1 - \sum_{j \in N_i} \beta_{ij}[\tau]$; we can rewrite (42) as

$$\begin{aligned} \bar{i}_i^c &= \beta_{ii}[\tau] \bar{i}_i + \sum_{j \in N_i} \beta_{ij}[\tau] \bar{i}_j, \\ \bar{I}_i^c &= \beta_{ii}[\tau] \bar{I}_i + \sum_{j \in N_i} \beta_{ij}[\tau] \bar{I}_j. \end{aligned} \quad (43)$$

Then we have

$$\begin{aligned} \bar{x}_i^c &= (\bar{I}_i^c)^{-1} \bar{i}_i^c = (\bar{I}_i^c)^{-1} \left[\beta_{ii}[\tau] \bar{i}_i + \sum_{j \in N_i} \beta_{ij}[\tau] \bar{i}_j \right] \\ &= W_{ii} \bar{x}_i + \sum_{j \in N_i} W_{ij} \bar{x}_j. \end{aligned} \quad (44)$$

Here we use $\bar{i}_i = \bar{I}_i \bar{x}_i$, $i \in J_i$ and $W_{il} = \beta_{il}[\tau] (\bar{I}_i^c)^{-1} \bar{I}_l$, $l \in J_i$. Then we finally get

$$\begin{aligned} \hat{x}_i &= \bar{x}_i^c + M_i (y^i - S^i \bar{x}_i^c) \\ &= W_{ii} \bar{x}_i + M_i (y^i - S^i W_{ii} \bar{x}_i) + (I - M_i S^i) \sum_{j \in N_i} W_{ij} \bar{x}_j \end{aligned}$$

Initialization (for node i):
 $\bar{i}_i = \bar{i}(0)$ $\bar{I}_i = \bar{I}(0)$
 $\tau = 1$ $\tau_p = \tau + T_p$

Loop {Local iteration on node i }

(1) Consensus update
 $\bar{i}_i^c = \bar{i}_i + \sum_{j \in N_i} \beta_{ij}[\tau](\bar{i}_j - \bar{i}_i)$
 $\bar{I}_i^c = \bar{I}_i + \sum_{j \in N_i} \beta_{ij}[\tau](\bar{I}_j - \bar{I}_i)$
 $y^i = \sum_{j \in J_i} H_j^T R_j^{-1} z_j$ $S^i = \sum_{j \in J_i} H_j^T R_j^{-1} H_j$ $\tau \leftarrow \tau + 1$

(2) If new observations are taken then the information consensus estimate are computed
 $\hat{i}_i^c = \bar{i}_i^c + y^i = \bar{i}_i + y^i + \sum_{j \in N_i} \beta_{ij}[\tau](\bar{i}_j - \bar{i}_i)$
 $\hat{I}_i = \bar{I}_i^c + S^i = \bar{I}_i + S^i + \sum_{j \in N_i} \beta_{ij}[\tau](\bar{I}_j - \bar{I}_i)$

(3) If time for a predication step (i.e., $\tau = \tau_p$) then prediction step
 $\bar{I}_i^+ = (A\hat{I}_i^{-1}A^T + BQB^T)^{-1}$
 $\bar{i}_i^+ = \bar{I}_i^+ A\hat{I}_i^{-1}\hat{i}_i$
 $\tau_p = \tau + T_p$

End Loop

ALGORITHM 3: Information consensus filter.

$$= \left((\bar{I}_i^c)^{-1} \bar{I}_i \right) \bar{x}_i + M_i \left[y^i - S^i \left((\bar{I}_i^c)^{-1} \bar{I}_i \right) \bar{x}_i \right] \\ + (I - M_i S^i) \sum_{j \in N_i} \beta_{ij}[\tau] \left[(\bar{I}_i^c)^{-1} \bar{I}_j \bar{x}_j - (\bar{I}_i^c)^{-1} \bar{I}_i \bar{x}_i \right],$$

$$M_i = (\hat{I}_i)^{-1} = \left[\bar{I}_i + S^i + \sum_{j \in N_i} \beta_{ij}[\tau] (\bar{I}_j - \bar{I}_i) \right]^{-1}, \quad (45)$$

where the error covariance matrix M_i has been improved by a consensus term compared with $M_i = (\hat{I}_i)^{-1} = [\bar{I}_i + S^i]^{-1}$ in KCF, and the state estimate \hat{x}_i is also corrected by a factor $((\bar{I}_i^c)^{-1} \bar{I}_i)$ compared with the \hat{x}_i in KCF.

5.4. The Optimization of Consensus Weights in ICF. The weights $\beta_{ij}[\tau]$ are important parameters of ICF and we can use the metropolis weights or the maximum-degree weights to determine it. In fact, the more reasonable approach is to choose different weights according to the fused local information I_i and i_i . Here, a nice scheme to optimize the consensus weights is proposed base on the following objective function:

$$F_i = \alpha_i^1 \frac{\text{tr} \left((\bar{I}_i^c)^{-1} \right)}{\text{tr} \left((\bar{I}_i)^{-1} \right)} \\ + \alpha_i^2 \frac{\| \bar{i}_i^c - \bar{i}_{i,av}^c \|^2 + \sum_{j \in N_i} \| \bar{i}_j^c - \bar{i}_{i,av}^c \|^2}{\sum_{j \in J_i} \| \bar{i}_j - \bar{i}_{i,av} \|^2}, \quad (46)$$

where

$$\bar{i}_{i,av}^c = \frac{1}{1 + d_i} \left(\bar{i}_i^c + \sum_{j \in N_i} \bar{i}_j^c \right),$$

$$\bar{i}_{i,av} = \frac{1}{1 + d_i} \left(\bar{i}_i + \sum_{j \in N_i} \bar{i}_j \right), \quad (47)$$

α_i^1, α_i^2 are the weight coefficients, and $\alpha_i^1 + \alpha_i^2 = 1$ ($0 < \alpha_i^1 < 1, 0 < \alpha_i^2 < 1$).

The first term in the objective function F_i is used to assess the prior estimate error covariance of node i after fusing the local information of its neighbors, and the second term is used to assess the consensus of the fused local information vector in node i and the prior estimates of its neighbors. Base on the objective function F_i , the consensus weights optimization problem can be described as

$$\beta_i^* = \arg \min_{\beta_i} F_i \\ \text{s.t. } \beta_{ij} \geq 0, \quad (i, j) \in E[\tau] \\ \beta_{ij} = 0, \quad (i, j) \notin E[\tau] \\ \|\beta_i\|_1 = 1, \quad (48)$$

where $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]$. To solve the optimization problem (48), we only need the local information of node i and that of its neighbors. We refer to this method as weights optimized information consensus filter (WO-ICF).

6. Numerical Simulations

Let the linear system under consideration be represented by a second-order discrete time-varying model:

$$x(k+1) = Ax(k) + Bw, \quad (49)$$

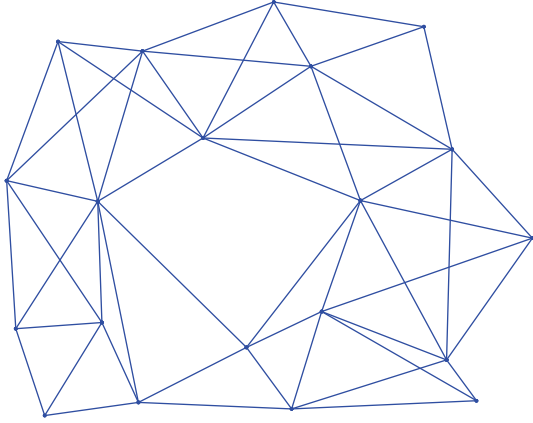


FIGURE 2: A sensor network with 20 nodes and 51 links.

where $A = I_2 + \delta A_0 + (\delta^2/2)A_0^2 + (\delta^3/6)A_0^3$ with $A_0 = 2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\delta = 0.015$ and $B = \delta B_0$ with $B_0 = I_2$, $Q = 25I_2$. The initial conditions are $x_0 = [15, -10]$, $P_0 = 20I_2$. A sensor network with 20 randomly located nodes is used in this experiment (see Figure 2). The local observation matrix for sensor i is $H_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, and the local observation noise covariance is $R_i = 100I_2$ for $i \leq 10$ otherwise $R_i = 3000I_2$.

Define the averaged estimation error $E(k)$ and the averaged consistency estimation error $D(k)$ as the algorithm performance metrics, which can be computed as follows:

$$E(k) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i(k) - x(k))^T (\hat{x}_i(k) - x(k))}, \quad (50)$$

$$D(k) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i(k) - \hat{x}_{av}(k))^T (\hat{x}_i(k) - \hat{x}_{av}(k))},$$

where $\hat{x}_{av}(k) = (1/n) \sum_{i=1}^n \hat{x}_i(k)$ is the averaged estimation of state.

Figure 3 demonstrates the averaged estimation error using different algorithms. We can see that the ICF and the WO-ICF behave in a similar manner (with comparable performances), and the averaged estimation accuracy in ICF and WO-ICF is improved compared to KCF. After 50 iterations, their performances are very close to CKF; this is because the average consensus is achieved after constantly information exchanging, fusing, and filtering.

Figure 4 shows that our WO-ICF performs significantly better than both KCF and ICF, it was the fastest converged, and the consistency of estimates between different nodes in WO-ICF was improved by optimizing the consensus weights.

Figure 5 demonstrates the comparisons of different algorithms on the traces of the averaged estimation covariance matrices $\text{tr}((1/n) \sum_{i=1}^n M_i(k))$. A quick look at Figure 5 reveals that both ICF and our WO-ICF perform significantly better than KCF, of which the reason is that the information matrix weighted consensus strategy is adopted. Furthermore, by optimizing the consensus weights, the error covariance matrix M_i is improved significantly compared to ICF.

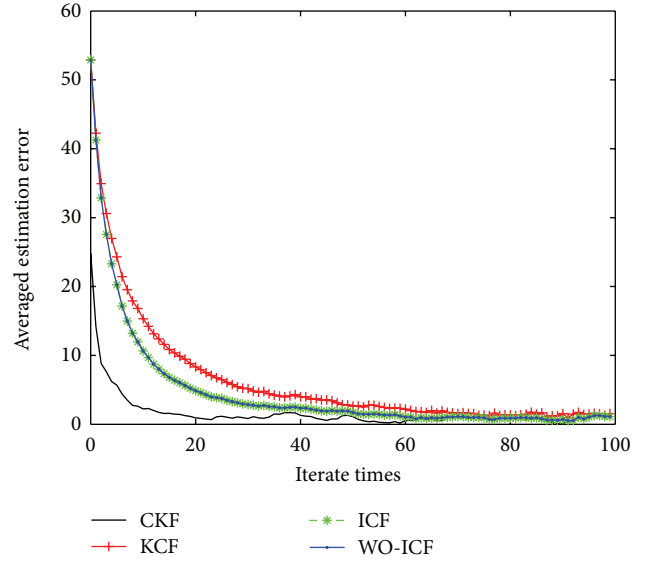


FIGURE 3: The averaged estimation errors of different algorithms.

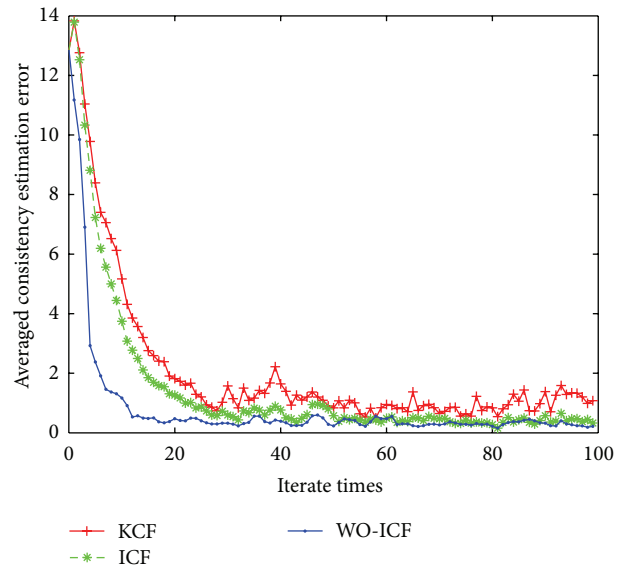


FIGURE 4: The averaged consistency estimation errors of different algorithms.

7. Conclusions

In this paper, a description about the existing distributed Filter with consensus strategies is presented, including Kalman consensus filter (KCF) and information consensus filter (ICF). In addition, an in-depth comparison between the KCF and the ICF is made. Based on the ICF, the weights optimized information consensus filter (WO-ICF) is proposed to optimize the consensus weights. Simulation shows that both ICF and WO-ICF perform better than KCF; they improve not only the state estimate but also the error covariance matrix, and the proposed WO-ICF achieves better consistency estimation performance than ICF. Compared with the existing consensus filter, our WO-ICF achieves the best performance and is closest to the optimal centralized performance.

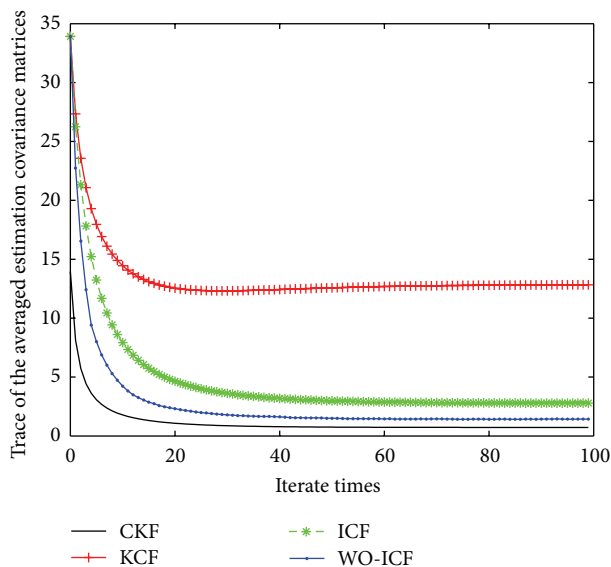


FIGURE 5: The traces of the averaged estimation covariance matrices of different algorithms.

Acknowledgments

This work is supported in part by NSFC (Natural Science Foundation of China) Projects 61202400, Natural Science Foundation of Zhejiang Q12F020078, Shaoxing Project of Science and Technology 2011A22013, and Wenzhou Project of Science and Technology H20090054, S20100029, and H20100095.

References

- [1] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann, 1997.
- [2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice Hall, Englewood Cliffs, NJ, USA, 1979.
- [3] A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere, New York, NY, USA, 1975.
- [4] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed kalman filtering in sensor networks with quantifiable performance," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 133–139, April 2005.
- [5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [6] R. Olfati Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the American Control Conference*, vol. 2, pp. 951–956, June 2003.
- [7] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 8179–8184, December 2005.
- [8] D. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proceedings of the 16th International Federation of Automatic Control World Congress*, Prague, Czech, 2005.
- [9] 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.
- [10] R. Olfati-Saber, "Kalman-Consensus filter: optimality, stability, and performance," in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference (CDC/CCC '09)*, pp. 7036–7042, Shanghai, China, December 2009.
- [11] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 5492–5498, December 2007.
- [12] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.
- [13] S. Utete and H. F. Durrant-Whyte, "Reliability in decentralized data fusion networks," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '94)*, pp. 215–221, October 1994.
- [14] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.
- [15] A. G. O. Mutambara, *Decentralized Estimation and Control For Multisensor Systems*, CRC Press, Boca Raton, Fla, USA, 1998.
- [16] D. W. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *Proceedings of the American Control Conference (ACC '09)*, pp. 1882–1887, Hyatt Regency Riverfront, St. Louis, Mo, USA, June 2009.
- [17] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [18] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [19] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 168–176, April 2006.
- [20] D. B. Kingston and R. W. Beard, "Discrete-time average-consensus under switching network topologies," in *Proceedings of the American Control Conference*, pp. 3551–3556, June 2006.
- [21] D. W. Casbeer and R. W. Beard, "Multi-static radar target tracking using information consensus filters," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Ill, USA, August 2009.