


Article

Distributed Kalman Filtering Based on the Non-Repeated Diffusion Strategy

Xiaoyu Zhang ¹ and Yan Shen ^{2,*} 

¹ College of Artificial Intelligence, Nankai University, Tianjin 300350, China; zhangxiaoyu@nankai.edu.cn

² College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China

* Correspondence: q23357320@hrbeu.edu.cn; Tel.: +86-1730-228-6361

Received: 6 November 2020; Accepted: 30 November 2020; Published: 3 December 2020

Abstract: Estimation accuracy is the core performance index of sensor networks. In this study, a kind of distributed Kalman filter based on the non-repeated diffusion strategy is proposed in order to improve the estimation accuracy of sensor networks. The algorithm is applied to the state estimation of distributed sensor networks. In this sensor network, each node only exchanges information with adjacent nodes. Compared with existing diffusion-based distributed Kalman filters, the algorithm in this study improves the estimation accuracy of the networks. Meanwhile, a single-target tracking simulation is performed to analyze and verify the performance of the algorithm. Finally, by discussion, it is proved that the algorithm exhibits good all-round performance, not only regarding estimation accuracy.

Keywords: distributed Kalman filter; diffusion strategy; sensor networks; data fusion

1. Introduction

Sensor detection is undergoing a transition from an independent style to a cooperative style, and sensor networks have found increasing numbers of applications in areas such as in the Internet of Things [1], environmental monitoring [2], cooperative radar detection [3] and autonomous driving [4]. According to the types of communication topologies, sensor networks are classified into centralized networks and distributed networks [5]. A centralized network needs a stable central node with excellent communication performance, which is not yet realistic in large-scale networks. Therefore, a large number of researchers are focusing on distributed sensor networks [6,7].

Multiple nodes in a distributed sensor network can simultaneously detect and obtain target state information with noise. Each node communicates with its adjacent nodes. By using the distributed estimation algorithm, each node obtains a global and more accurate estimation than a single sensor. The distributed sensor network does not need to use a high-performance central node and is much more easily extended to a large-scale network. In short, this kind of sensor network has high stability and robustness and a low sensor cost [8]. Therefore, it is necessary to study distributed sensor networks. There are two main existing distributed Kalman filters, namely the consensus-based algorithm and diffusion-based algorithm.

The idea of consensus-based algorithms is to obtain the global average measure or the global average information vector and information matrix by performing the average consensus algorithm [9,10]. Olfatih–Saber first proposed a consensus-based distributed algorithm, called the Kalman consensus filter [11], which can obtain the mean value of global information vectors and the information matrix after infinite iterations and obtain the same global optimality estimations as centralized filters. In [12], the authors address the problem of the consensus of the information vector and information matrix, which requires infinite iterations to achieve the optimal estimation.

To handle the correlation problem of estimations, in [13], the authors propose the information consensus filter (ICF), which fuses the prior information with low weights and the measured information of adjacent nodes. To handle local unobservability, a finite-time consensus-based distributed estimator is proposed in [14], which is based on the max-consensus technique. In [15,16], the authors propose a gossip-based algorithm that adopts a random communication strategy to achieve consensus. In contrast to consensus-based algorithms, a lower bandwidth can also guarantee the consensus of gossip-based algorithms with the cost of a slow convergence speed. Consensus-based algorithms have the same estimation accuracy as the centralized filter, whereas the global consensus is only achieved asymptotically.

Diffusion-based algorithms exchange intermediate estimations between the neighborhoods of each node and calculate the final estimation by a convex combination of information from neighbors. In [17,18], the authors obtain the sum of global measurement information in finite communication cycles through the diffusion of measurement information and propose a finite-time distributed Kalman filter (FT-DKF). In [19], the authors extend the algorithm proposed in [17] to cyclic graphs. The work presented in [20] was the first to propose a distributed estimation fusion filter based on diffusion. The accuracy of diffusion-based algorithms depends on the selection of convex combination coefficients. For this reason, in [21], the authors discuss the optimal selection of convex combination coefficients and formulate a constrained optimization problem. In [22], the authors propose the cost-effective diffusion Kalman filter (CE-DKF), which diffuses the information of state estimations and estimation covariances and improves the performance. The work presented in [23] was the first to apply the covariance intersection (CI) method to deal with the diffusion-based distributed Kalman filtering problem and address the problem of the system noise correlation between sensors. In [24], the authors use the local CI to improve local estimation performance. In [25], the authors proposed a new distributed Kalman filter that avoids the diffusion of raw data and maintains estimation accuracy by resorting to maximum posterior probability state estimation. In [26], the results of [25] are improved upon and the distributed hybrid information fusion (DHIF) algorithm is proposed. The partial diffusion Kalman filter (PDKF) is proposed for the state estimation of linear dynamic systems in [27]. By only sending partial estimation vectors at each iteration, the PDKF algorithm reduces the number of internode communication. In [28], the authors study the performance of PDKF for networks with noises. In summary, diffusion-based algorithms improve the consensus convergence speed, but the estimation accuracy is reduced at the same time.

From the above review, we can see that the diffusion-based algorithms have some advantages and some defects. Therefore, in this paper, we aim to design a kind of diffusion distributed Kalman filter for application to a sensor network. This sensor network is time-invariant and strongly connected. Each sensor node has its own information and the information from adjacent sensors. By using the distributed Kalman filter, the estimations of all nodes contain the global information and reflect the real state of the target after finite communication iterations.

First, a non-repeated diffusion strategy is introduced. The usual diffusion strategy is bi-directional; this means that the information diffuses back in some way. The non-repeated diffusion strategy involves removing the information that is received from a node before sending all messages to this node so that the information is not diffused back. This strategy makes information diffuse only in one direction, even in an undirected graph. Second, we apply the non-repeated diffusion strategy to the diffusion-based distributed Kalman filter and propose a new distributed Kalman filter in which the coefficients of convex combinations are obtained by CI. The introduction of the non-repeated diffusion strategy prevents sensors from fusing the same estimation repeatedly, which may lead to some estimations being too highly weighted and the final estimation results to be biased toward these estimations. Compared with existing distributed Kalman filters, the algorithm in this study has a lower error of estimations. Besides, we show that the algorithm also has good performance in other aspects, including communication bandwidth requirements, communication frequency requirements, applicability to different topologies and robustness to local unobservability. A trace simulation example

is provided to verify the performance of the distributed Kalman filter based on the non-repeated diffusion strategy.

The rest of the paper is organized as follows. In Section 2, the system model and a diffusion distributed Kalman filter based on the non-repeated diffusion strategy are introduced. The result of the algorithm is shown in Section 3, which verifies the effectiveness of the algorithm. Section 4 discusses this result and describes the application scenarios of this algorithm. The conclusion of this paper is summarized in Section 5.

2. Diffusion Distributed Kalman Filtering Algorithms

2.1. System Model

In this study, the sensor network is represented by a graph $G = (V, E)$, where $V = \{1, 2, 3, \dots, N\}$ is the set of sensor nodes (where N is the number of nodes in the sensor network and E is the set of communication channels (edges) between sensors). A graph is called an undirected graph if it consists of undirected edges. If an undirected edge $(i, j) \in E$ exists, node i and node j are neighbors and can communicate with each other. We use N_i to represent the set of adjacent nodes connected to the node i . When there is a path between each pair of different nodes i and j , the graph is called a connected graph. A path is called a cyclic path if this path starts from node A to node B and returns to node A through node C . If a graph is connected and has no cyclic path, it is called an acyclic graph (tree graph). We use d to represent the diameter of the graph G , which is the length of the longest path. d_{ij} denotes the length of the shortest path between nodes i and j .

For the distributed estimation problem under this sensor network, we consider the following target state model

$$x(t+1) = Ax(t) + \Gamma\omega(t), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $A \in \mathbb{R}^{n \times n}$ is the state transition matrix, and $\Gamma \in \mathbb{R}^{n \times h}$ is a noise coefficient matrix. $\omega(t) \in \mathbb{R}^h$ is a system noise vector which is a zero-mean Gaussian white noise. The covariance matrix of $\omega(t)$ is $Q > 0$. Each node of the sensor network can observe the target, and the measurement equation of each node is

$$y_i(t) = H_i x(t) + v_i(t), \quad (2)$$

where $y_i(t) \in \mathbb{R}^m$ is the measurement vector of sensor i , and $H \in \mathbb{R}^{m \times n}$ is the observation matrix. $v_i(t) \in \mathbb{R}^m$ is a measurement noise vector, which is a zero-mean Gaussian white noise. The covariance matrix of $v_i(t)$ is $R_i > 0$. The covariance matrix of system noise $\omega(t)$ and measurement noise $v_i(t)$, $\forall i \in V$ is

$$\mathbb{E}[\omega(t)v_i(t)] = O_{h \times m}. \quad (3)$$

The covariance matrix of measurement noise $v_i(t)$, $\forall i \in V$ and measurement noise $v_j(t)$, $\forall j \in V$ is

$$\mathbb{E}[v_i(t)v_j(t)] = O_{m \times m}. \quad (4)$$

To facilitate the expression, we introduce the following notations: $y_{i,t}$ is the measurement of target from the node i at time t , $\hat{x}_{i,t|t-1}$ is the state prediction at time t based on measurements by the sensor i up to time $t-1$, $P_{i,t|t-1}$ is an estimation error covariance matrix of $\hat{x}_{i,t|t-1}$. $\hat{x}_{i,t|t}$ and $P_{i,t|t}$ are the same as $\hat{x}_{i,t|t-1}$ and $P_{i,t|t-1}$.

2.2. A New Diffusion Distributed Kalman Filter

In this subsection, we propose a new diffusion distributed Kalman filter. First, we introduce the seminal diffusion distributed Kalman filter [20], which is summarized in Algorithm 1. $\hat{x}_{i,t|t}^{loc}$ is the estimation with local information. $N_i + i$ is a set composed of set N_i and node i , which contains node i and all adjacent nodes of node i .

Algorithm 1: The seminal diffusion distributed Kalman filter.

For the node $i \in V$ and node $j \in N_i$,

Initialize with:

$$\hat{x}_{i,0|0} = \mathbb{E}x_0,$$

$$P_{i,0|0} = \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T];$$

Local update:

$$\hat{x}_{i,t|t-1} = A\hat{x}_{i,t-1|t-1},$$

$$P_{i,t|t-1} = AP_{i,t-1|t-1}A^T + \Gamma Q \Gamma^T,$$

$$(P_{i,t|t})^{-1} = (P_{i,t|t-1})^{-1} + H_i^T R_i^{-1} H_i,$$

$$\hat{x}_{i,t|t}^{loc} = \hat{x}_{i,t|t-1} + P_{i,t|t} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1});$$

Communication and fusion update:

Send $\hat{x}_{i,t|t}^{loc}$ to adjacent node i ,

$$\hat{x}_{i,t|t} = \sum_{k \in N_i + i} w_k \hat{x}_{k,t|t}^{loc},$$

where

$$\sum_{k \in N_i + i} w_k = 1, 0 \leq w_k \leq 1.$$

The algorithm fuses the estimation of adjacent nodes, increases the information exchange and improves the estimation accuracy compared with the Kalman filter of a single sensor. The shortcoming of this algorithm is that it does not deal with the estimation error covariance. Therefore, $P_{i,t-1|t-1}$ in Algorithm 1 is not the accurate estimation error covariance of $\hat{x}_{i,t-1|t-1}$ in Algorithm 1, and the estimation accuracy of Algorithm 1 is reduced. In this work, we diffuse and fuse the estimation error covariance to optimize Algorithm 1.

In general, due to the process noise and complex sensor interactions, the cross-covariance between different estimations is unknown. In the case of unknown cross-covariance, we can use CI [25], the ellipsoidal intersection (EI) method [29] and the information sharing principle [30], etc. The information sharing principle requires the distribution of the covariance of global estimation to each node according to a certain proportion, which leads to a great burden of communication. From Algorithm 1, we know that the diffusion-based distributed Kalman filtering algorithm is a convex combination of estimations of adjacent nodes. Similarly, the EI method and CI method obtain more accurate estimations by adjusting the convex combination coefficients of multiple estimates. From this perspective, we can apply the EI method and CI method to extend the diffusion-based distributed filtering algorithm. However, the EI method has some disadvantages compared with the CI method. First, the EI method can provide a more accurate estimation, but its result does not result in consistent estimation. Second, the choice of whether to use EI method is an engineering problem. Third, the calculation process of the EI method is complex, which results in it being inconvenient to describe the algorithm proposed in this paper. Therefore, we use the CI method to fuse estimations, as described in Algorithm 2.

In Algorithm 2, $P_{i,t|t}^{loc}$ is the estimation error covariance of $\hat{x}_{i,t|t}^{loc}$ and w_i , $i \in V$ are weight coefficients of CI that minimize the following equation:

$$\min_{w_i \geq 0, \sum_{i \in V} w_i = 1} \left[\text{tr} \left(\sum_{i \in V} w_i P_{i,t|t}^{-1} \right) \right] \quad (5)$$

with $\text{tr}(\cdot)$ being the trace function. Equation (5) is a nonlinear optimization problem whose computational cost is usually expensive and unaffordable for sensors. The weight can be set as the mean to reduce computational cost.

Algorithm 2: A centralized Kalman filter based on the covariance intersection (CI) method.

For the node $i \in V$,

Initialize with:

$$\hat{x}_{i,0|0}^{\text{loc}} = \mathbb{E}x_0,$$

$$P_{i,0|0}^{\text{loc}} = \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T];$$

Local update:

$$P_{i,t|t-1}^{\text{loc}} = AP_{i,t-1|t-1}^{\text{loc}}A^T + \Gamma Q \Gamma^T,$$

$$\hat{x}_{i,t|t-1}^{\text{loc}} = A\hat{x}_{i,t-1|t-1}^{\text{loc}},$$

$$\hat{x}_{i,t|t}^{\text{loc}} = \hat{x}_{i,t|t-1}^{\text{loc}} + H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1}^{\text{loc}}),$$

$$(P_{i,t|t}^{\text{loc}})^{-1} = (P_{i,t|t-1}^{\text{loc}})^{-1} + H_i^T R_i^{-1} H_i;$$

Fusion update:

$$P_{t|t}^{-1} = \sum_{i \in V} w_i (P_{i,t|t}^{\text{loc}})^{-1},$$

$$\hat{x}_{t|t} = P_{t|t} \sum_{i \in V} w_i (P_{i,t|t}^{\text{loc}})^{-1} \hat{x}_{i,t|t}^{\text{loc}},$$

where w_j is calculated by Equation (5).

The error covariance of estimation calculated by CI is approximate, and so the result of Algorithm 2 is also approximate, whereas CI provides a conservative result which guarantees that the accuracy in a specific direction is not lower than that of any node in this direction. The estimation of Algorithm 2 and the optimal estimation satisfy the following equation:

$$\hat{x}_{t|t} \approx \mathbb{E}(x_t | y_{i,0}, y_{i,1}, \dots, y_{i,t}, i \in V). \quad (6)$$

Applying Algorithm 2 to Algorithm 1, we obtain a diffusion distributed Kalman filter that fuses estimation and its error covariance concurrently, which is presented in Algorithm 3.

Algorithm 3: A diffusion distributed Kalman filter with the CI method.

For the node $i \in V$ and node $j \in N_i$,

Initialize with:

$$\hat{x}_{i,0|0} = \mathbb{E}x_0,$$

$$P_{i,0|0} = \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T];$$

Local update:

$$\hat{x}_{i,t|t-1} = A\hat{x}_{i,t-1|t-1},$$

$$P_{i,t|t-1} = AP_{i,t-1|t-1}A^T + \Gamma Q \Gamma^T,$$

$$(P_{i,t|t}^{\text{loc}})^{-1} = (P_{i,t|t-1})^{-1} + H_i^T R_i^{-1} H_i,$$

$$\hat{x}_{i,t|t}^{\text{loc}} = \hat{x}_{i,t|t-1} + P_{i,t|t} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1});$$

Communication and fusion update:

Send $\hat{x}_{j,t|t}^{\text{loc}}$ and $P_{j,t|t}^{\text{loc}}$ to adjacent node i ,

$$P_{i,t|t}^{-1} = \sum_{k \in N_i+i} w_k (P_{k,t|t}^{\text{loc}})^{-1},$$

$$\hat{x}_{i,t|t} = P_{i,t|t} \left[\sum_{k \in N_i+i} w_k (P_{k,t|t}^{\text{loc}})^{-1} \hat{x}_{k,t|t}^{\text{loc}} \right], \quad (7)$$

where w_k is calculated by Equation (5).

In Algorithm 3, $\hat{x}_{i,t|t}^{\text{loc}}$ is calculated from $\hat{x}_{i,t-1|t-1}$ by local update. $\hat{x}_{i,t-1|t-1}$ fuses information of $\hat{x}_{j,t-1|t-1}^{\text{loc}}$. Therefore, $\hat{x}_{i,t|t}^{\text{loc}}$ also contains information of $\hat{x}_{j,t-1|t-1}^{\text{loc}}$ and $\hat{x}_{i,t|t}^{\text{loc}}$ is correlative to $\hat{x}_{j,t-1|t-1}^{\text{loc}}$. Since $\hat{x}_{j,t|t}^{\text{loc}}$ is also correlative to $\hat{x}_{j,t-1|t-1}^{\text{loc}}$, $\hat{x}_{i,t|t}^{\text{loc}}$ is correlative to $\hat{x}_{j,t|t}^{\text{loc}}$. In Equation (7), Algorithm 3 fuses $\hat{x}_{i,t|t}^{\text{loc}}$ and $\hat{x}_{j,t|t}^{\text{loc}}$ and omits their correlation. In order to improve the accuracy of Algorithm 3, we separate

diffusion update from local update, and propose a diffusion distributed Kalman filter based on CI in Algorithm 4.

Algorithm 4: A diffusion distributed Kalman filter separating diffusion update.

For the node $i \in V$ and node $j \in N_i$,

Initialize with:

$$\begin{aligned}\hat{x}_{i,0|0}^{\text{loc}} &= \mathbb{E}x_0, \\ P_{i,0|0}^{\text{loc}} &= \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T], \\ \hat{x}_{i,0|0} &= \mathbb{E}x_0, \\ P_{i,0|0} &= \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T];\end{aligned}$$

Local update:

$$\begin{aligned}\hat{x}_{i,t|t-1}^{\text{loc}} &= A\hat{x}_{i,t-1|t-1}^{\text{loc}}, \\ P_{i,t|t-1}^{\text{loc}} &= AP_{i,t-1|t-1}^{\text{loc}}A^T + \Gamma Q \Gamma^T, \\ (P_{i,t|t}^{\text{loc}})^{-1} &= (P_{i,t|t-1}^{\text{loc}})^{-1} + H_i^T R_i^{-1} H_i, \\ \hat{x}_{i,t|t}^{\text{loc}} &= \hat{x}_{i,t|t-1}^{\text{loc}} + P_{i,t|t}^{\text{loc}} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1}^{\text{loc}});\end{aligned}$$

Diffusion incremental Update:

$$\begin{aligned}\hat{x}_{i,t|t-1} &= A\hat{x}_{i,t-1|t-1}, \\ P_{i,t|t-1} &= AP_{i,t-1|t-1}A^T + \Gamma Q \Gamma^T, \\ (P_{i,t|t}^{\text{diffusion}})^{-1} &= P_{i,t|t-1}^{-1} + H_i^T R_i^{-1} H_i, \\ \hat{x}_{i,t|t}^{\text{diffusion}} &= \hat{x}_{i,t|t-1} + P_{i,t|t}^{\text{diffusion}} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1});\end{aligned}$$

Communication and fusion update:

$$\begin{aligned}\text{Send } \hat{x}_{j,t|t}^{\text{diffusion}} \text{ and } P_{j,t|t}^{\text{diffusion}} \text{ to adjacent node } i, \\ P_{i,t|t}^{-1} &= \sum_{j \in N_i} w_j (P_{j,t|t}^{\text{diffusion}})^{-1} + w_i (P_{i,t|t}^{\text{loc}})^{-1}, \\ \hat{x}_{i,t|t} &= P_{i,t|t} [\sum_{j \in N_i} w_j (P_{j,t|t}^{\text{diffusion}})^{-1} \hat{x}_{j,t|t}^{\text{diffusion}} + w_i (P_{i,t|t}^{\text{loc}})^{-1} \hat{x}_{i,t|t}^{\text{loc}}],\end{aligned}$$

where w_j and w_i is calculated by Equation (5).

In Algorithm 4, each node requires a local update and diffusion incremental update. The results $\hat{x}_{i,t|t}^{\text{diffusion}}$ and $P_{i,t|t}^{\text{diffusion}}$ of the incremental diffusion update are diffused to adjacent nodes. The results $\hat{x}_{i,t|t}$ and $P_{i,t|t}^{\text{loc}}$ of the local update are used to fuse with the received information. Compared with Algorithm 3, Algorithm 4 uses the local information to fuse with the received information in the fusion process instead of using the fusion results of the last moment to fuse with the received information; this solves the correlation problem caused by fusing the received information of the last moment with the received information at the current moment in Algorithm 3 at every moment.

2.3. Distributed Kalman Filter Based on the Non-Repeated Diffusion Strategy

In this subsection, we proposed a distributed Kalman filter based on the non-repeated diffusion strategy and analyze its performance. First, we introduce the non-repeated diffusion strategy in Algorithm 5. Each node in Algorithm 5 has local information $I_{i,t}^{\text{loc}}$ at each moment. $I_{i,t}$ is all of the information of node i at time t and is calculated by Equation (9). $I_{i \rightarrow i_1,t}$ is the information that is sent from node i to node i_1 at time t and is calculated by Equation (8). In this strategy, the diffusion information for each adjacent node is calculated. The diffusion information $I_{i \rightarrow i_1,t}$ is calculated by subtracting the received information from node i_1 at the last moment from all information at the last moment. The strategy avoids the sent information from a node returning to this node again.

Algorithm 5: The non-repeated diffusion strategy.

For the node $i \in V$ and node $i_1 \in N_i$,

Initialize information of node i at time t with $I_{i,t}^{\text{loc}}$;

Initialize with:

$$I_{i_1 \rightarrow i,0} = 0;$$

Diffusion update:

$$I_{i \rightarrow i_1,t} = \sum_{i_2 \in N_{i_1-i_1}} I_{i_2 \rightarrow i_1,t-1} + I_{i_1,t}^{\text{loc}}; \quad (8)$$

Communication and fusion information:

Send $I_{i_1 \rightarrow i,t}$ to adjacent node i ,

$$I_{i,t} = \sum_{i_1 \in N_i} I_{i_1 \rightarrow i,t} + I_{i,t}^{\text{loc}}. \quad (9)$$

Assumption 1. The sensor network is an acyclic graph.

Theorem 1. Let Assumption 1 hold. Running Algorithm 5, no node of the sensor network will ever receive its own information.

Proof of Theorem 1. There is a node i in an acyclic graph. Substituting Equation (8) into Equation (9), we obtain

$$I_{i,t} = \sum_{i_1 \in N_i} \left(\sum_{i_2 \in N_{i_1-i_1}} I_{i_2 \rightarrow i_1,t-1} + I_{i_1,t}^{\text{loc}} \right) + I_{i,t}^{\text{loc}}. \quad (10)$$

Substituting Equation (8) into Equation (10) many times, all information of node i at time t can be presented as

$$I_{i,t} = \sum_{i_1 \in N_i} \left(\sum_{i_2 \in N_{i_1-i_1}} \cdots \left(\sum_{i_n \in N_{i_{n-1}-i_{n-2}}} I_{i_n \rightarrow i_{n-1},t-(n-1)} + I_{i_{n-1},t-(n-2)}^{\text{loc}} \right) \cdots + I_{i_1,t}^{\text{loc}} \right) + I_{i,t}^{\text{loc}}, \quad (11)$$

where $n \in \mathbb{N}^*$. The sets of adjacent nodes of any two points in an acyclic graph do not produce an intersection set (except for each other); otherwise, there will be a cyclic path through the intersection set. Thus, there is no intersection set among $N_{i_{n-1}} - i_{n-2}$, $n \in \mathbb{N}^*$ and the element of $N_{i_{n-1}} - i_{n-2}$ will constantly decrease as n increases. There is an m that means that $N_{i_{n-1}} - i_{n-2} = \emptyset$, $n \geq m$. At this moment, all information of node i is reduced as

$$I_{i,t} = I_{i,t}^{\text{loc}} + \sum_{i_1 \in N_i} I_{i_1,t}^{\text{loc}} + \sum_{i_2 \in N_{i_1}} I_{i_2,t}^{\text{loc}} + \cdots + \sum_{i_m \in N_{i_{m-1}}} I_{i_m,t-(m-1)}^{\text{loc}} \quad (12)$$

where $I_{i,t}^{\text{loc}}$ is the information that node i already has. Therefore, the node i never receives its own information from other nodes. \square

We introduce the non-repeated diffusion strategy into Algorithm 4 and propose a new diffusion distributed Kalman filter, which is the core of this paper and is presented in Algorithm 6. In Algorithm 6, $\hat{x}_{i \rightarrow j,t|t}$ and $P_{i \rightarrow j,t|t}$ are the estimation and its error covariance diffused from node i to j .

Algorithm 6: A distributed Kalman filter based on the non-repeated diffusion strategy.

For the node $i \in V$ and node $j \in N_j$.

Initialize with:

$$\begin{aligned}\hat{x}_{i,0|0}^{\text{loc}} &= \mathbb{E}x_0, P_{i,0|0}^{\text{loc}} = \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T], \\ \hat{x}_{j \rightarrow i,0|0} &= 0, P_{j \rightarrow i,0|0} = \mathbb{E}[(x_0 - \mathbb{E}x_0)(x_0 - \mathbb{E}x_0)^T];\end{aligned}$$

Local update:

$$\begin{aligned}\hat{x}_{i,t|t-1}^{\text{loc}} &= A\hat{x}_{i,t-1|t-1}^{\text{loc}} \\ P_{i,t|t-1}^{\text{loc}} &= AP_{i,t-1|t-1}^{\text{loc}}A^T + \Gamma Q \Gamma^T, \\ (P_{i,t|t}^{\text{loc}})^{-1} &= (P_{i,t-1|t-1}^{\text{loc}})^{-1} + H_i^T R_i^{-1} H_i, \\ \hat{x}_{i,t|t}^{\text{loc}} &= \hat{x}_{i,t|t-1}^{\text{loc}} + P_{i,t|t}^{\text{loc}} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i,t|t-1}^{\text{loc}});\end{aligned}$$

Diffusion incremental update:

$$P_{i \rightarrow j,t}^{-1} = \sum_{g \in N_{i-j}} w_{j,g \rightarrow i} P_{g \rightarrow i,t-1|t-1}^{-1} + w_{j,i \rightarrow i} (P_{i,t-1|t-1}^{\text{loc}})^{-1}, \quad (13)$$

$$\hat{x}_{i \rightarrow j,t} = P_{i \rightarrow j,t} [\sum_{g \in N_{i-j}} w_{j,g \rightarrow i} P_{g \rightarrow i,t|t}^{-1} \hat{x}_{g \rightarrow i,t-1|t-1} + w_{j,i \rightarrow i} (P_{i,t-1|t-1}^{\text{loc}})^{-1} \hat{x}_{i,t-1|t-1}^{\text{loc}}], \quad (14)$$

$$\hat{x}_{i \rightarrow j,t|t-1} = A\hat{x}_{i \rightarrow j,t},$$

$$P_{i \rightarrow j,t|t-1} = AP_{i \rightarrow j,t}A^T + \Gamma Q \Gamma^T,$$

$$P_{i \rightarrow j,t|t}^{-1} = P_{i \rightarrow j,t|t-1}^{-1} + H_i^T R_i^{-1} H_i,$$

$$\hat{x}_{i \rightarrow j,t|t} = \hat{x}_{i \rightarrow j,t|t-1} + P_{i \rightarrow j,t|t} H_i^T R_i^{-1} (y_{i,t} - H_i \hat{x}_{i \rightarrow j,t|t-1});$$

Communication and fusion update:

Send $\hat{x}_{j \rightarrow i,t|t}$ and $P_{j \rightarrow i,t|t}$ to adjacent node i ,

$$P_{i,t|t}^{-1} = \sum_{j \in N_i} w_{j \rightarrow i} P_{j \rightarrow i,t|t}^{-1} + w_{i \rightarrow i} (P_{i,t|t}^{\text{loc}})^{-1},$$

$$\hat{x}_{i,t|t} = P_{i,t|t} [\sum_{j \in N_i} w_{j \rightarrow i} P_{j \rightarrow i,t|t}^{-1} \hat{x}_{j \rightarrow i,t|t} + w_{i \rightarrow i} (P_{i,t|t}^{\text{loc}})^{-1} \hat{x}_{i,t|t}^{\text{loc}}],$$

where w_j is calculated by Equation (19).

Theorem 2. Let Assumption 1 hold. The estimations calculated by Algorithm 6 fuse the global information and reflect the real state of the target after a finite number of communications.

Proof of Theorem 2. The step of the fusion update in Algorithm 4 combines information from adjacent nodes of node i linearly. In the step of the diffusion update, all information of node i is diffused to all adjacent nodes. Therefore, adjacent nodes of node i can receive the information from the other adjacent nodes of node i . In every communication, the information of each node is diffused to the next node. Algorithm 6 adds two steps—Equations (13) and (14)—which are derived from the non-repeated diffusion strategy of Algorithm 5. From Theorem 1, we know that Algorithm 6 realizes the prerequisite that each node never receives its own information. Since the information is diffused to the next node and each node does not receive its own information, the information is always diffused in one direction. Each node can receive the information from other nodes besides its neighbors. The amount of received information depends on the number of other nodes within a certain distance. Due to applying CI to the fusing of the information, a conservative estimation is obtained that is the same as Equation (6). The estimation of node i is

$$\hat{x}_{i,t|t} \approx \mathbb{E}(x_t | y_{l,1}, \dots, y_{l,t_{il}}, l \in V'), \quad (15)$$

where

$$t_{il} = t - d_{il} + 1, \quad (16)$$

and V' is a set of nodes, whose measures can be received by node i . $\{y_{l,1}, \dots, y_{l,t_{il}}, l \in V'\}$ represents the measurements of nodes in V' , and t_{il} denotes the maximum moment before which the measurement information from node l can be received by node i .

Assume that the two nodes with the longest distance in the sensor network are node i and node j ($d_{ij} = d$); with d communications after time t_1 (at time $t = t_1 + d - 1$), we obtain

$$t_{il} = t_1 + d - d_{il}, \forall l \in V. \quad (17)$$

For node j , there is

$$t_{ij} = t_1. \quad (18)$$

At this moment, node i receives the measure y_{j,t_1} . Each node can receive the estimates and covariances of all nodes in the sensor network because the two nodes with the longest distance can communicate. Due to the fusing of the estimations and covariances of all nodes by CI, each node obtains an estimation that includes the global information and reflects the real state of target after d communications. \square

$w_{j,g \rightarrow i}$, $w_{j,i \rightarrow i}$, $w_{j \rightarrow i}$ and $w_{i \rightarrow i}$ in Algorithm 6 are the weight coefficients of CI, which minimize the following equation (take $w_{j \rightarrow i}$, $w_{i \rightarrow i}$ as an example):

$$\min_{w_{j \rightarrow i} \geq 0, w_{i \rightarrow i} \geq 0, \sum_{j \in N_i} w_{j \rightarrow i} + w_{i \rightarrow i} = 1} \left[\text{tr} \left(\sum_{j \in N_i} w_{j \rightarrow i} \tilde{\zeta}_{j \rightarrow i} + w_{i \rightarrow i} (P_{i,t-1|t-1}^{\text{loc}})^{-1} \right) \right]. \quad (19)$$

Equation (19) is a nonlinear optimization problem, as with Equation (5), whose computational cost is also expensive and unaffordable for sensors. The weight can be set as the mean to reduce computational cost.

3. Results

In this section, an example is simulated to show the efficiency of Algorithm 6, and the comparisons with CE-DKF [22] and DHIF [26] are also shown. CE-DKF and DHIF are two existing prominent diffusion-based distributed filter algorithms.

Specifically, we consider a sensor network of 20 nodes whose topological structure is shown in Figure 1. Each node can detect the target and communicate bi-directionally with its adjacent nodes. The linear dynamical system of Equation (1) is considered, where the state vector $x_t = [\alpha_t \ \beta_t \ \dot{\alpha}_t \ \dot{\beta}_t]^T$ consists of position α_t, β_t and speed $\dot{\alpha}_t, \dot{\beta}_t$ from horizontal and vertical directions, respectively. The unit is meters. The initial state is $[0 \ 0 \ 20 \ 20]^T$. The acceleration of the target is modeled as the system noise $\omega(t) = [\ddot{\alpha}_t \ \ddot{\beta}_t]^T$ whose mean is $[0 \ 0]^T$ and variance is Q . Here, we consider the case where Q is equal to

$$Q_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}^T$$

or

$$Q_2 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}^T.$$

The state transition matrix is

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the noise coefficient matrix is

$$\Gamma = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The sensor measurement model is Equation (2), where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

the mean of $v_i(t)$ is $[0 \ 0 \ 0 \ 0]^T$, and the variance of $v_i(t)$ is

$$R_i = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T, \forall i \in V.$$

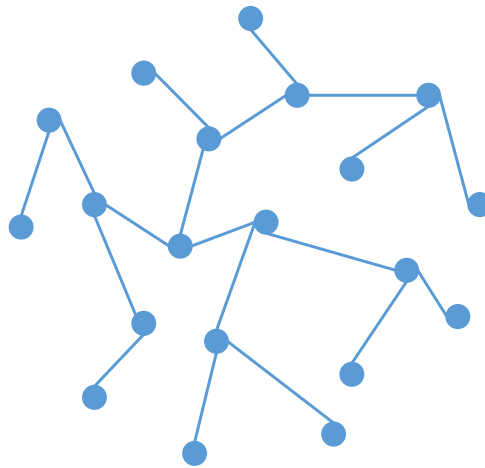


Figure 1. A sensor network with 20 nodes.

Next, we perform a simulation 2000 times to compare Algorithm 6 with other distributed filtering algorithms. The fusion weights of both Algorithm 6 and DHIF are mean values.

Figures 2 and 3 show the estimation error of the algorithms for state α_t . The equation to calculate the error is

$$\varepsilon = \sqrt{\frac{1}{N} \sum_{i \in V} (\alpha_{i,t|t} - \alpha_t)^2}. \quad (20)$$

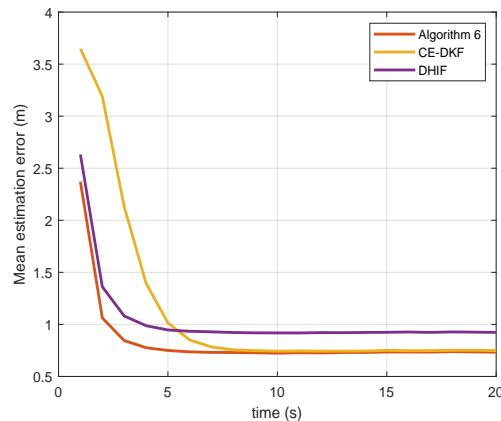


Figure 2. Mean estimation errors of different estimation algorithms ($Q = Q_1$). CE-DKF: cost-effective diffusion Kalman filter; DHIF: distributed hybrid information fusion.

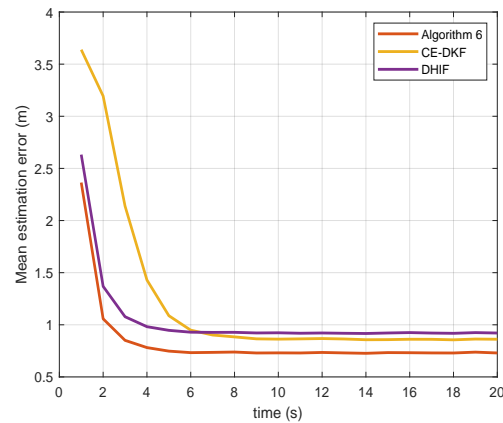


Figure 3. Mean estimation errors of different estimation algorithms ($Q = Q_2$).

The data of Figures 2 and 3 are collated in Table 1.

Table 1. Error performance comparison of different algorithms.

Algorithms	Steady State Error (m)		Error Reduction of Algorithm 6 (%)		Error Convergence Time (s)	
	Q_1	Q_2	Q_1	Q_2	Q_1	Q_2
Algorithm 6	0.7301	0.7297	0	0	7	7
CE-DKF	0.7476	0.8640	2.34	15.54	9	10
DHIF	0.9214	0.9233	20.76	20.97	8	9

The error reduction of Algorithm 6 is the percentage of error that Algorithm 6 reduces compared to other algorithms. The error convergence time is the earliest time when the error reaches a steady state. From Table 1, we see that Algorithm 6 reduces the estimation error by up to 20.97% and reaches a steady state faster than CE-DKF and DHIF. For the case in which the noise covariance matrix Q is large, the performance of Algorithm 6 is more prominent. In the case of a small Q , Algorithm 6 only reduces the error by 2.34% compared with CE-DKF. Because the maneuverability of target is small in this case, the advantages of Algorithm 6 are not obvious.

Figures 4 and 5 show the standard deviations (SDs) of the estimations of sensors for state α_t , which represents the degree of consensus of estimations from all nodes in the sensor network and is calculated by

$$\Phi = \sqrt{\frac{1}{N} \sum_{i \in V} \left(\alpha_{i,k|k} - \frac{1}{N} \sum_{j \in V} \alpha_{j,k|k} \right)^2}. \quad (21)$$

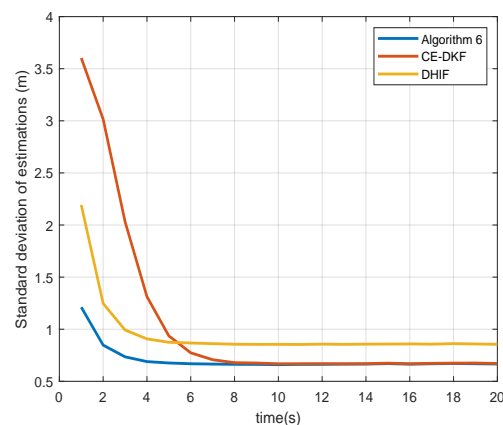


Figure 4. Standard deviation of estimations ($Q = Q_1$).

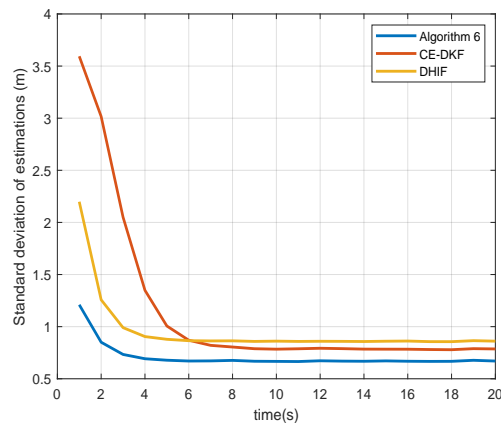


Figure 5. Standard deviation of estimations ($Q = Q_2$).

The data in Figures 4 and 5 are collated in Table 2.

Table 2. Consensus performance comparison of different algorithms.

Algorithms	Steady State SD (m)		SD Reduction of Algorithm 6 (%)		Convergence Time of SD (s)	
	Q_1	Q_2	Q_1	Q_2	Q_1	Q_2
Algorithm 6	0.6655	0.6702	0	0	7	7
CE-DKF	0.6710	0.7882	0.81	14.97	10	9
DHIF	0.8570	0.8631	22.34	22.35	8	8

The SD reduction of Algorithm 6 is the percentage by which Algorithm 6 reduces the SD compared to other algorithms. The convergence time of SD is the earliest time when SD reaches a steady state. From Table 1, we see that Algorithm 6 reduces the SD of estimation by up to 22.34% and reaches a steady state faster than CE-DKF and DHIF. This means that Algorithm 6 can ensure that all nodes in the sensor network have a higher degree of consensus. In the case of a small Q , Algorithm 6 only reduces SD by 0.81% compared with CE-DKF. The reason for this is the same as above: the error advantage of Algorithm 6 is small in the case of a small Q .

In each communication, each sensor applying Algorithm 6 only needs to send its own estimation $\hat{x}(t | t) \in \mathbb{R}^n$ and the estimation error covariance $P \in \mathbb{R}^{n \times n}$. Therefore, the communication traffic is $n \times n + n$ in terms of the number of transmitted digits. We give the communication bandwidth and frequency requirement of some popular algorithms in Table 3. The communication frequency in this study means the number of communications with an adjacent node required for each node to complete estimation. In addition to CE-DKF and DHIF, we compare the communication requirements of other two consensus-based algorithms. From Table 3, we can see that the communication bandwidth and frequency requirement of Algorithm 6 is the lowest (or one of the lowest).

Table 3. Communication requirements of different algorithms. FT-DKF: finite-time distributed Kalman filter; ICF: information consensus filter.

Algorithms	Bandwidth Requirement	Frequency Requirement (Hz)
Algorithm 6	$n \times n + n$	1
ICF [13]	$n \times n + n$	d
FT-DKF [18]	$n \times n + n$	d
CE-DKF	$n \times n + n$	1
DHIF	$2 \times (n \times n + n)$	1

4. Discussion

In this section, we discuss the result of Algorithm 6, including its estimation accuracy, degree of consensus, communication requirement, applicability to different topologies and robustness to local unobservability.

4.1. Estimation Accuracy

The reason why Algorithm 6 has better estimation accuracy than other diffusion-based algorithms is that Algorithm 6 applies the non-repeated diffusion strategy, which eliminates repeated information and reduces redundancy. In addition, Algorithm 6 improves the efficiency of information exchange due to non-redundant information exchange, meaning that it can converge to a steady-state value faster than other diffusion-based algorithms. The accuracy advantage of Algorithm 6 is small in the case of a small Q ; this means that Algorithm 6 is good at estimating unknown models and maneuvering states. The only disadvantage is that Algorithm 6 compresses the fusion weight in the process of communication, which causes the fusion weight of Algorithm 6 to deviate and reduces the accuracy. This is a limitation of Algorithm 6. Future work will present a better weighting strategy.

4.2. Degree of Consensus

In sensor network estimation, it is very important that the estimation of all nodes is consistent. According to Section 3, Algorithm 6 has a lower estimation variance and a higher degree of consensus. It can be seen from Tables 1 and 2 that the variance has a certain correlation with estimation error. Therefore, it can be considered that the reduction of error reduces the estimation dispersion range and the variance of Algorithm 6. Similarly, because the error converges faster, the consensus convergence rate is also faster. The advantage of the degree of consensus is small in the case of a small Q ; this also means that Algorithm 6 is good at estimating unknown models and maneuvering states.

4.3. Communication Requirement

There are a large number of low-cost sensor networks with low communication bandwidths. For this kind of sensor network, the communication bandwidth requirement of the distributed filter algorithm has to be considered. From Section 3, we know that the communication traffic of Algorithm 6 is competitive against other distributed algorithms.

As with the communication bandwidth, communication frequency is limited for low-cost sensor networks. Some algorithms must satisfy a certain number of communications in order to receive global information and maintain a certain accuracy. When the number of communications is 1 Hz, Algorithm 6 can still guarantee that every node obtains global information.

When the communication frequency is less than 1 Hz, Algorithm 6 can still guarantee that every node obtains global information by combining measurements. In short, at a low communication frequency, as long as the communication frequency is greater than 0 Hz, Algorithm 6 can obtain better accuracy than some algorithms that must run in a certain communication frequency. This communication frequency requirement is also the advantage of diffusion-based algorithms, and thus we apply the diffusion-based algorithm and optimize it.

4.4. Applicability to Different Topologies

Theorem 2 assumes that the topology of the sensor network is an acyclic graph (tree graph). In fact, Algorithm 6 can also be applied to cyclic graphs. For sensor networks with a cyclic path, we can apply the spanning tree algorithm to convert a cyclic graph to an acyclic graph before applying Algorithm 6. Without doubt, this increases the complexity of Algorithm 6, which is also one of the limitations of Algorithm 6. Next, we can study this aspect and time-varying sensor networks.

4.5. Robustness to Local Unobservability

In some cases, some nodes miss the target but other nodes detect the target. The weight $w_{j \rightarrow i}$ of the nodes that miss the target can be considered to be 0. At this moment, Assumption 1 still holds. Since Theorem 2 is independent of the node weight, Algorithm 6 still guarantees that Theorem 2 holds. Therefore, Algorithm 6 is robust to local unobservability and is suitable for wider sensor networks.

5. Conclusions

In this study, a distributed Kalman filter is proposed based on the non-repetition diffusion strategy. It operates in a fully distributed mode and does not need to obtain information about the sensor network, which gives it excellent expansibility. The core of the algorithm lies in the non-repeated diffusion strategy, which optimizes the diffusion-based distributed Kalman filtering algorithms. Compared with existing distributed Kalman filtering algorithms, the algorithm proposed in this study has a lower estimation error. Besides, the algorithm also shows outstanding performance in terms of the degree of consensus, communication bandwidth requirement, communication frequency requirement, applicability to different topologies and robustness to local unobservability. The algorithm shows excellent performance in the tracking of maneuvering targets and the state estimation of low-cost sensor networks. By the simulation of a single target trace, we verify the performance of the algorithm. Future research topics include the weighting strategy and time-varying sensor networks.

Author Contributions: Conceptualization, X.Z.; methodology, Y.S.; software, Y.S.; validation, X.Z.; formal analysis, Y.S.; resources, X.Z.; data curation, X.Z.; writing—original draft preparation, Y.S.; writing—review and editing, X.Z.; visualization, Y.S.; supervision, X.Z.; project administration, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICF	Information consensus filter
FT-DKF	Finite-time distributed Kalman filter
CE-DKF	Cost-effective diffusion Kalman filter
CI	Covariance intersection
DHIF	Distributed hybrid information fusion
PDKF	Partial diffusion Kalman filter
EI	Ellipsoidal intersection
SD	Standard deviation

References

- Oztemel, E.; Gursev, S. Literature review of Industry 4.0 and related technologies. *J. Intell. Manuf.* **2020**, *31*, 127–182. [[CrossRef](#)]
- Silva, B.; Fisher, R.M.; Kumar, A.; Hancke, G.P. Experimental Link Quality Characterization of Wireless Sensor Networks for Underground Monitoring. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1099–1110. [[CrossRef](#)]
- Javadi, S.H.; Farina, A. Radar networks: A review of features and challenges. *Inf. Fusion* **2020**, *61*, 48–55. [[CrossRef](#)]
- Cai, P.; Wang, S.; Sun, Y.; Liu, M. Probabilistic End-to-End Vehicle Navigation in Complex Dynamic Environments With Multimodal Sensor Fusion. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4218–4224. [[CrossRef](#)]
- Sun, S.; Lin, H.; Ma, J.; Li, X. Multi-sensor distributed fusion estimation with applications in networked systems: A review paper. *Inf. Fusion* **2017**, *38*, 122–134. [[CrossRef](#)]
- Kumar, K.A.S.; Ramakrishnan, K.R.; Rathna, G.N. Distributed sigma point information filters for target tracking in camera networks. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications, Tokyo, Japan, 18–22 May 2015; pp. 373–377.

7. Paola, D.D.; Petitti, A.; Rizzo, A. Distributed Kalman filtering via node selection in heterogeneous sensor networks. *Int. J. Syst. Sci.* **2015**, *46*, 2572–2583. [[CrossRef](#)]
8. He, S.; Shin, H.S.; Xu, S.; Tsourdos, A. Distributed estimation over a low-cost sensor network: A Review of state-of-the-art. *Inf. Fusion* **2020**, *54*, 21–43. [[CrossRef](#)]
9. Saber, R.O.; Murray, R.M. Consensus protocols for networks of dynamic agents. In Proceedings of the 2003 American Control Conference, Denver, CO, USA, 4–6 June 2003; Volume 2, pp. 951–956.
10. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533. [[CrossRef](#)]
11. Olfati-Saber, R. Distributed kalman filter with embedded consensus filters. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 12–15 December 2005; pp. 8179–8184.
12. Olfati-Saber, R. Distributed Kalman filtering for sensor networks. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.
13. Kamal, A.T.; Farrell, J.A.; Roy-Chowdhury, A.K. Information weighted consensus. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control, Maui, HI, USA, 10–13 December 2012; pp. 2732–2737.
14. Liu, P.; Tian, Y.; Zhang, Y. Distributed Kalman Filtering With Finite-Time Max-Consensus Protocol. *IEEE Access* **2018**, *6*, 10795–10802. [[CrossRef](#)]
15. Franceschelli, M.; Giua, A.; Seatzu, C. Consensus on the average on arbitrary strongly connected digraphs based on broadcast gossip algorithms. In Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy, 24–26 September 2009. [[CrossRef](#)]
16. Ma, K.; Wu, S.; Wei, Y.; Zhang, W. Gossip-Based Distributed Tracking in Networks of Heterogeneous Agents. *IEEE Commun. Lett.* **2017**, *21*, 801–804. [[CrossRef](#)]
17. Tai, X.; Lin, Z.; Fu, M.; Sun, Y. A new distributed state estimation technique for power networks. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 3338–3343.
18. Wu, Z.; Fu, M.; Xu, Y.; Lu, R. A distributed Kalman filtering algorithm with fast finite-time convergence for sensor networks. *Automatica* **2018**, *95*, 63–72. [[CrossRef](#)]
19. Sui, T.; Marelli, D.; Fu, M.; Lu, R. Accuracy analysis for distributed weighted least-squares estimation in finite steps and loopy networks. *Automatica* **2018**, *97*, 82–91. [[CrossRef](#)]
20. Cattivelli, F.S.; Sayed, A.H. Diffusion Strategies for Distributed Kalman Filtering and Smoothing. *IEEE Trans. Autom. Control* **2010**, *55*, 2069–2084. [[CrossRef](#)]
21. Cattivelli, F.; Sayed, A.H. Diffusion distributed Kalman filtering with adaptive weights. In Proceedings of the 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–4 November 2009; pp. 908–912.
22. Talebi, S.P.; Kanna, S.; Xia, Y.; Mandic, D.P. Cost-effective diffusion Kalman filtering with implicit measurement exchanges. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, Orleans, LA, USA, 5–9 March 2017; pp. 4411–4415.
23. Hlinka, O.; Slučiak, O.; Hlawatsch, F.; Rupp, M. Distributed data fusion using iterative covariance intersection. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 1861–1865.
24. Zhang, Y.; Wang, C.; Li, N.; Chambers, J. Diffusion Kalman filter based on local estimate exchanges. In Proceedings of the 2015 IEEE International Conference on Digital Signal Processing, Singapore, 21–24 July 2015; pp. 828–832.
25. Wang, G.; Li, N.; Zhang, Y. Diffusion distributed Kalman filter over sensor networks without exchanging raw measurements. *Signal Process.* **2017**, *132*, 1–7. [[CrossRef](#)]
26. Wang, S.; Ren, W. On the Convergence Conditions of Distributed Dynamic State Estimation Using Sensor Networks: A Unified Framework. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1300–1316. [[CrossRef](#)]
27. Vahidpour, V.; Rastegarnia, A.; Khalili, A.; Sanei, S. Partial Diffusion Kalman Filtering for Distributed State Estimation in Multiagent Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3839–3846. [[CrossRef](#)]
28. Vahidpour, V.; Rastegarnia, A.; Latifi, M.; Khalili, A.; Sanei, S. Performance Analysis of Distributed Kalman Filtering With Partial Diffusion Over Noisy Network. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1767–1782. [[CrossRef](#)]

29. Sijs, J.; Lazar, M.; Bosch, P.P.J.V.D. State fusion with unknown correlation: Ellipsoidal intersection. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 3992–3997.
30. Carlson, N.A. Federated square root filter for decentralized parallel processors. *IEEE Trans. Aerosp. Electron. Syst.* **1990**, *26*, 517–525. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).