

Article

Iterative Diffusion-Based Distributed Cubature Gaussian Mixture Filter for Multisensor Estimation

Bin Jia ¹, Tao Sun ² and Ming Xin ^{2,*}

¹ Intelligent Fusion Technology, Germantown, MD 20876, USA; binjiaqm@gmail.com

² Department of Mechanical and Aerospace Engineering, University of Missouri, Columbia, MO 65211, USA; tspg5@mail.missouri.edu

* Correspondence: xin@missouri.edu; Tel.: +1-573-882-7933

Academic Editor: Xue-Bo Jin

Received: 30 July 2016; Accepted: 15 October 2016; Published: 20 October 2016

Abstract: In this paper, a distributed cubature Gaussian mixture filter (DCGMF) based on an iterative diffusion strategy (DCGMF-ID) is proposed for multisensor estimation and information fusion. The uncertainties are represented as Gaussian mixtures at each sensor node. A high-degree cubature Kalman filter provides accurate estimation of each Gaussian mixture component. An iterative diffusion scheme is utilized to fuse the mean and covariance of each Gaussian component obtained from each sensor node. The DCGMF-ID extends the conventional diffusion-based fusion strategy by using multiple iterative information exchanges among neighboring sensor nodes. The convergence property of the iterative diffusion is analyzed. In addition, it is shown that the convergence of the iterative diffusion can be interpreted from the information-theoretic perspective as minimization of the Kullback–Leibler divergence. The performance of the DCGMF-ID is compared with the DCGMF based on the average consensus (DCGMF-AC) and the DCGMF based on the iterative covariance intersection (DCGMF-ICI) via a maneuvering target-tracking problem using multiple sensors. The simulation results show that the DCGMF-ID has better performance than the DCGMF based on noniterative diffusion, which validates the benefit of iterative information exchanges. In addition, the DCGMF-ID outperforms the DCGMF-ICI and DCGMF-AC when the number of iterations is limited.

Keywords: sensor networks; distributed estimation; Gaussian mixture; diffusion

1. Introduction

With the rapid progress of the sensing and computing technologies, multiple sensors have been widely used in estimation applications, such as target tracking, wireless sensor networks, guidance and navigation, and environmental monitoring. Effective information fusion from multiple sensors is of utmost importance. It can be done in a centralized or distributed manner. For the centralized fusion, the information obtained by all sensors is collected and processed by the central node. This approach enables the global solution but requires a large amount of power and resources in communication and computation. The failure or delay on the central node may significantly degrade the estimation performance. For the distributed estimation, the information at each sensor node is processed locally and then fused to establish the global information by well-designed distributed fusion algorithms using only the local information. In contrast to the centralized estimation, the distributed estimation offers a number of advantages, such as scalability, robustness to single point of failure, low communication load, and low operation cost.

When the estimation is processed at each local sensor, it is a regular filtering problem, which has been intensively researched for decades. In many practical estimation problems, the system dynamics and measurement equations are nonlinear and the uncertainties or noises are non-Gaussian. To address

this challenging filtering problem, Gaussian mixture-based filters [1] and sequential Monte Carlo-based filters [2] are two classes of widely used approaches. The rationale behind the Gaussian mixture-based filters is that any probability density function (pdf) can be approximated by the summation of a finite number of Gaussian distributions. The Monte Carlo-based filters or particle filters use a large number of particles to represent the pdf. Although some solutions have been proposed to alleviate the curse of the dimensionality problem for application of particle filters in high-dimensional problems, the computation complexity is still prohibitive. Therefore, from the computation efficiency perspective in the sensor network setting, the Gaussian mixture filter is a better alternative and will be used in this paper for multiple sensor estimation. The mean and covariance of each Gaussian component are predicted and updated using the cubature Kalman filtering (CKF) algorithm [3,4]. The fifth-degree CKF [4] is used because it is more accurate than the conventional third-degree CKF in [3] and other well-known nonlinear Gaussian filters such as the extended Kalman filter (EKF) [5] and the unscented Kalman filter (UKF) [6], which is a third-degree Gaussian filter as well.

After the local estimation is obtained at each sensor node, information fusion of the estimates from multiple sensors is conducted using the distributed estimation algorithm. Distributed estimation has been a research subject of considerable interest in the past few years [7–17]. Olfati-Saber [7,8] first addressed the distributed estimation problem by reducing it to two average consensus filters, one for weighted measurement and the other for information form of the covariance matrix. Because each sensor node only communicates with its immediate neighbors, the average consensus strategy is effective to obtain the average of each node's initial value. In each iteration, each node updates its state by weighting its prior state and its neighbors' prior states. When the number of iterations approaches infinity, average consensus can be achieved. In the consensus-based distributed estimation framework, certain requirement on the network topology is usually necessary. In [9,10], information from an individual node is propagated through the entire network via a new information-weighted consensus scheme. Although each node has limited observability of the states, even including naive agents (not having measurement), the proposed information-weighted consensus filter for distributed maximum a posteriori parameter estimation and state estimation is capable of obtaining a final estimate comparable to that obtained from the centralized filter. However, it only considered the scenario that all local estimates and measurement errors are independent or uncorrelated. Sun et al. [11] proposed a batch covariance intersection technique combined with average consensus algorithms to address the correlation issue. But, the Gaussian assumption is made on all estimation processes. It may be inadequate for highly nonlinear systems and/or non-Gaussian systems. On the other hand, due to the constraints on energy and communication frequency, a large number of iterations in consensus algorithms are not feasible in practice, especially for the systems in which the time interval between two consecutive measurements is very small.

Diffusion strategies for distributed estimation proposed in [12] overcome the disadvantage of excessive energy and communication requirements in the average consensus-based estimation. There are two steps between consecutive filtering cycle in the diffusion algorithm: incremental and diffusion. The incremental step runs a local filtering at each node with a regular time update and multiple measurement updates by incrementally incorporating measurements from every neighboring node. The diffusion step computes the ultimate fused estimate by convex combination of all estimates from the present node and its neighbors. Each node only communicates with its direct neighbors twice in each filtering cycle. The first communication collects the innovation information from its neighbors. The second communication exchanges the state estimate among neighbors from the incremental step to do the diffusion update. The estimate obtained through the diffusion strategy has been proved unbiased for linear systems. The paper [12] also provides the mean, mean square, and convergence analysis and shows that the estimate is stable under the assumption that the state space model is time invariant and each local system (joint measurement model of one node and its immediate neighbors) is detectable and stabilizable. As long as the individual node satisfies the assumption, this diffusion strategy does not have any requirement for the network topology. Diffusion recursive least-squares

(RLS) algorithm was developed in [13] to deal with the distributed estimation problem and achieved the performance close to the global solution. It does not require transmission or inversion of matrices and, therefore, reduces computational complexity. It was shown that the distributed solution is asymptotically unbiased and stable if the regressors are zero-mean and temporally independent, and the inverse of covariance matrices at different time indexes can be replaced by its expected value. A diffusion least-mean-squares (LMS) algorithm was proposed in [14] with two versions: adapt-then-combine and combine-then-adapt. Mean and mean square performance were analyzed. Besides, the scheme of optimizing the diffusion LMS weights was discussed. The work of [15] extended the work in [12] by using the covariance intersection to yield a consistent estimate and relaxing the assumption made in [12]. It only requires partial local uniform observability rather than all local systems' observability assumed in [12]. The case of no local uniform observability was discussed in [15] as well but relied on the consensus filter. Hlinka et al. [16] proposed the distributed estimation scheme using the iterative covariance intersection (ICI). Like the consensus strategy, the ICI needs recursive update of each node's state and covariance until they converge. Each iteration can guarantee a consistent estimate. However, the ICI does not include the incremental update as the diffusion does.

Most of the aforementioned work assumes a linear dynamic process and measurement with Gaussian noise or initial uncertainty with Gaussian pdf. For highly nonlinear dynamic systems with non-Gaussian statistics, the performance of those distributed estimation methods may degrade. In this paper, we propose a new distributed Gaussian mixture filtering based on an iterative diffusion strategy to handle the distributed nonlinear estimation. There is limited literature on the distributed Gaussian mixture filtering. In [17], the likelihood consensus strategy was used in the design of a distributed Gaussian mixture filter in a sensor network that was not fully connected. Unlike the original consensus-based distributed estimation, the Gaussian mixture weight cannot be updated through the consensus filter directly since it needs to evaluate a product term of the likelihood function. By the natural logarithm transformation, the product term is transformed to a summation to which the consensus algorithm can be applied. The contributions of the proposed approach in this paper are: (1) a new distributed Gaussian mixture filtering framework with an embedded cubature rule can more accurately handle nonlinear and non-Gaussian distributed estimation problems; (2) the iterative diffusion strategy provides better fusion performance than the original diffusion method, the average consensus, and the ICI; (3) it does not need intensive communications as required in the consensus-based estimation; (4) the convergence analysis and information theoretic interpretation of the proposed approach are given.

The remainder of this paper is organized as follows. In Section 2, a centralized cubature Gaussian mixture filter is introduced. The distributed cubature Gaussian mixture filter using iterative diffusion is proposed in Section 3. In Section 4, the performance demonstration via a target-tracking problem is presented. Concluding remarks are given in Section 5.

2. Centralized Cubature Gaussian Mixture Filter

Consider a class of nonlinear discrete-time dynamic systems described by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \quad (1)$$

$$\mathbf{y}_{k,j} = \mathbf{h}_j(\mathbf{x}_k) + \mathbf{n}_{k,j} \quad (2)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector and $\mathbf{y}_{k,j} \in \mathbb{R}^m$ is the measurement by the j th sensor where the subscript "j" denotes the sensor index. \mathbf{v}_{k-1} and $\mathbf{n}_{k,j}$ are the process noise and measurement noise, respectively, and their probability density functions (pdf) are represented by the Gaussian mixtures (GM) $p(\mathbf{v}_k) = \sum_{p=1}^{N_p} \alpha^p N(\mathbf{v}_k^p; \bar{\mathbf{v}}_k^p, \mathbf{Q}_k^p)$ and $p(\mathbf{n}_{k,j}) = \sum_{q=1}^{N_q} \alpha_j^q N(\mathbf{n}_{k,j}^q; \bar{\mathbf{n}}_{k,j}^q, \mathbf{R}_{k,j}^q)$, respectively, where $N(\mathbf{n}_{k,j}^q; \bar{\mathbf{n}}_{k,j}^q, \mathbf{R}_{k,j}^q)$ denotes a normal distribution with mean $\bar{\mathbf{n}}_{k,j}^q$ and covariance $\mathbf{R}_{k,j}^q$, and α is the weight of the Gaussian component. The superscripts "p" and "q" denote the p th and q th component of the GM; " N_p " and " N_q "

denote the number of Gaussian components. Due to the non-Gaussian noise and nonlinear dynamics, the estimated state will have a non-Gaussian pdf, which can be modeled as the GM as well.

2.1. Cubature Gaussian Mixture Kalman Filter

Assume that the initial state pdf at the beginning of each filtering cycle can be represented by the GM $p(\mathbf{x}) = \sum_{l=1}^{N_l} \alpha^l N(\mathbf{x}; \hat{\mathbf{x}}^l, \mathbf{P}^l)$. In Figure 1, one cycle of the cubature Gaussian mixture filter (CGMF) is illustrated. The cubature Kalman filter (CKF) [3,4] runs on each component of the GM to predict and update the component's mean and covariance. The prediction step of the CKF is first used for each of the N_l GM components. Note that after the prediction step, there are $N_l \times N_p$ Gaussian components contributed by the GM of the initial state pdf and the GM of the process noise. After that, the update step of the CKF is used for each Gaussian component and leads to $N_l \times N_p \times N_q$ Gaussian components added by the GM of the measurement noise. It can be seen that the number of Gaussian components increases after each filtering cycle. To limit the computational complexity, the number of Gaussian components has to be reduced after the update step. In the following, the prediction step and the update step for each Gaussian component using the CKF framework [3,4] are introduced.

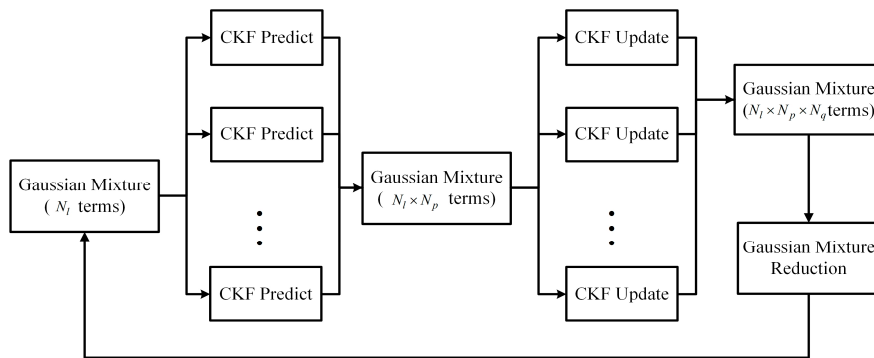


Figure 1. One filtering cycle of the cubature Gaussian mixture filter (CGMF).

2.1.1. Prediction Step

Given the initial estimate of the mean $\hat{\mathbf{x}}_{k-1|k-1}^l$ and covariance $\mathbf{P}_{k-1|k-1}^l$ at time $k-1$ for the l th Gaussian component, the predicted mean and covariance can be computed by the quadrature approximation [3,4]

$$\hat{\mathbf{x}}_{k|k-1}^{l,p} = \sum_{i=1}^{N_u} W_i f(\xi_{k-1,i}^l) + \bar{\mathbf{v}}_{k-1}^p \quad (3)$$

$$\mathbf{P}_{k|k-1}^{l,p} = \sum_{i=1}^{N_u} W_i \left[f(\xi_{k-1,i}^l) - (\hat{\mathbf{x}}_{k|k-1}^{l,p} - \bar{\mathbf{v}}_{k-1}^p) \right] \left[f(\xi_{k-1,i}^l) - (\hat{\mathbf{x}}_{k|k-1}^{l,p} - \bar{\mathbf{v}}_{k-1}^p) \right]^T + \mathbf{Q}_k^p \quad (4)$$

where N_u is the total number of cubature points, $l = 1, \dots, N_l$, $p = 1, \dots, N_p$; The superscript “ l,p ” denotes the value using the l th Gaussian component of the GM of the initial state pdf and the p th component of the GM of the process noise. $\bar{\mathbf{v}}_{k-1}^p$ is the mean of the p th Gaussian component of the GM representation of the process noise; $\xi_{k-1,i}^l$ is the transformed cubature point given by

$$\xi_{k-1,i}^l = \mathbf{S}_{k-1}^l \gamma_i + \hat{\mathbf{x}}_{k-1|k-1}^l, \quad \mathbf{P}_{k-1|k-1}^l = \mathbf{S}_{k-1}^l \left(\mathbf{S}_{k-1}^l \right)^T \quad (5)$$

The cubature points γ_i and weights W_i of the third-degree cubature rule [3] are given by

$$\gamma_i = \begin{cases} \sqrt{n} \mathbf{e}_i & i = 1, \dots, n \\ -\sqrt{n} \mathbf{e}_{i-n} & i = n+1, \dots, 2n \end{cases} \quad (6a)$$

$$W_i = 1 / (2n), \quad i = 1, \dots, 2n \quad (6b)$$

where \mathbf{e}_i is a unit vector with the i th element being 1. In this paper, the fifth-degree cubature rule [4] is also used to improve the estimation accuracy. The weights W_i and points γ_i of the fifth-degree rule are given by

$$W_1 = 2 / (n + 2) \quad (7a)$$

$$W_i = \frac{n^2 (7 - n)}{2 (n + 1)^2 (n + 2)^2}, \quad i = 2, \dots, 2n + 3 \quad (7b)$$

$$W_i = \frac{2 (n - 1)^2}{(n + 1)^2 (n + 2)^2}, \quad i = 2n + 4, \dots, n^2 + 3n + 3 \quad (7c)$$

$$\gamma_1 = \mathbf{0} \quad (8a)$$

$$\gamma_i = \sqrt{n + 2} \times \bar{\mathbf{s}}_{i-1}, \quad i = 2, \dots, n + 2 \quad (8b)$$

$$\gamma_i = -\sqrt{n + 2} \times \bar{\mathbf{s}}_{i-n-2}, \quad i = n + 3, \dots, 2n + 3 \quad (8c)$$

$$\gamma_i = \sqrt{n + 2} \times \tilde{\mathbf{s}}_{i-2n-3}, \quad i = 2n + 4, \dots, 2n + 3 + n(n + 1)/2 \quad (8d)$$

$$\gamma_i = -\sqrt{n + 2} \times \tilde{\mathbf{s}}_{i-(2n+3+n(n+1)/2)}, \quad i = 2n + 4 + n(n + 1)/2, \dots, n^2 + 3n + 3 \quad (8e)$$

where the points $\bar{\mathbf{s}}_i$ are given by

$$\bar{\mathbf{s}}_i = [p_{i,1}, p_{i,2}, \dots, p_{i,n}], \quad i = 1, 2, \dots, n + 1 \quad (9)$$

$$p_{i,j} \triangleq \begin{cases} -\sqrt{\frac{n+1}{n(n-j+2)(n-j+1)}} & j < i \\ \sqrt{\frac{(n+1)(n-i+1)}{n(n-i+2)}} & i = j \\ 0 & j > i \end{cases} \quad (10)$$

and

$$\{\tilde{\mathbf{s}}_i\} \triangleq \left\{ \sqrt{\frac{n}{2(n-1)}} (\bar{\mathbf{s}}_k + \bar{\mathbf{s}}_l) : k < l; k, l = 1, 2, \dots, n + 1 \right\} \quad (11)$$

2.1.2. Update Step

$$\hat{\mathbf{x}}_{k|k}^{l,p,q} = \hat{\mathbf{x}}_{k|k-1}^{l,p} + \mathbf{L}_k^{l,p,q} (\mathbf{y}_k - \mathbf{z}_k^{l,p,q}) \quad (12)$$

$$\mathbf{P}_{k|k}^{l,p,q} = \mathbf{P}_{k|k-1}^{l,p} - \mathbf{L}_k^{l,p,q} (\mathbf{P}_{xz}^{l,p,q})^T \quad (13)$$

$$\mathbf{L}_k^{l,p,q} = \mathbf{P}_{xz}^{l,p,q} (\mathbf{R}_k^q + \mathbf{P}_{zz}^{l,p,q})^{-1} \quad (14)$$

$$\mathbf{z}_k^{l,p,q} = \sum_{i=1}^{N_u} W_i \mathbf{h}(\tilde{\xi}_{k,i}^{l,p}) + \bar{\mathbf{n}}_k^q \quad (15)$$

$$\mathbf{P}_{xz}^{l,p,q} = \sum_{i=1}^{N_u} W_i \left[\tilde{\xi}_{k,i}^{l,p} - (\hat{\mathbf{x}}_{k|k-1}^{l,p} - \bar{\mathbf{v}}_k^p) \right] \left[\mathbf{h}(\tilde{\xi}_{k,i}^{l,p}) - (\mathbf{z}_k^{l,p,q} - \bar{\mathbf{n}}_k^q) \right]^T \quad (16)$$

$$\mathbf{P}_{zz}^{l,p,q} = \sum_{i=1}^{N_u} W_i \left[\mathbf{h}(\tilde{\xi}_{k,i}^{l,p}) - (\mathbf{z}_k^{l,p,q} - \bar{\mathbf{n}}_k^q) \right] \left[\mathbf{h}(\tilde{\xi}_{k,i}^{l,p}) - (\mathbf{z}_k^{l,p,q} - \bar{\mathbf{n}}_k^q) \right]^T \quad (17)$$

$\bar{\mathbf{n}}_k^q$ is the mean of the q th Gaussian component of the GM representation of the measurement noise; $\tilde{\xi}_{k,i}^{l,p}$ is the transformed cubature point given by

$$\tilde{\xi}_{k,i}^{l,p} = \tilde{\mathbf{S}}_k^{l,p} \gamma_i + \hat{\mathbf{x}}_{k|k-1}^{l,p}, \quad \mathbf{P}_{k|k-1}^{l,p} = \tilde{\mathbf{S}}_k^{l,p} \left(\tilde{\mathbf{S}}_k^{l,p} \right)^T \quad (18)$$

Remark 1: The weight for the Gaussian component $N(\mathbf{x}; \hat{\mathbf{x}}_{k|k}^{l,p,q}, \mathbf{P}_{k|k}^{l,p,q})$ is $\alpha^l \cdot \alpha^p \cdot \alpha^q$. The final GM can be represented by $\sum_{l=1}^{N_l} \sum_{p=1}^{N_p} \sum_{q=1}^{N_q} \alpha^l \alpha^p \alpha^q N(\mathbf{x}; \hat{\mathbf{x}}_{k|k}^{l,p,q}, \mathbf{P}_{k|k}^{l,p,q})$.

Note that the number of Gaussian components increases significantly as the time evolves. In order to avoid excessive computation load, some Gaussian components can be removed or merged. There are many GM reduction algorithms [18–20], such as pruning Gaussian components with negligible weights, joining near Gaussian components, and regeneration of GM via Kullback–Leibler approach. In this paper, near Gaussian components are joined to reduce the number of Gaussian components. The detailed description of this method is omitted since it is not the focus of this paper and can be seen in [20]. Note that to keep the estimation accuracy, the GM reduction procedure is not necessary if the number of Gaussian components is less than a specified threshold. For the convenience of implementing the diffusion update step in the proposed distributed estimation algorithm, the number of reduced Gaussian components at each sensor node is specified a priori to be the same.

2.2. Centralized Cubature Gaussian Mixture Filter

The centralized cubature Gaussian mixture filter (CCGMF) can be more conveniently expressed using the information filtering form. In the information filter, the information state and the information matrix of the Gaussian component with index l, p, q at time $k - 1$ are defined as $\hat{\mathbf{y}}_{k-1|k-1}^{l,p,q} = \left(\mathbf{P}_{k-1|k-1}^{l,p,q} \right)^{-1} \hat{\mathbf{x}}_{k-1|k-1}^{l,p,q}$ and $\mathbf{Y}_{k-1|k-1}^{l,p,q} = \left(\mathbf{P}_{k-1|k-1}^{l,p,q} \right)^{-1}$, respectively. The prediction of the information state and information matrix can be obtained via Equations (3) and (4). Using the information from multiple sensors, the information state and the information matrix can be updated by [4,21]

$$\hat{\mathbf{y}}_{k|k}^{l,p,q} = \hat{\mathbf{y}}_{k|k-1}^{l,p} + \sum_{j=1}^{N_{sn}} \mathbf{i}_{k,j}^{l,p,q} \quad (19)$$

$$\mathbf{Y}_{k|k}^{l,p,q} = \mathbf{Y}_{k|k-1}^{l,p} + \sum_{j=1}^{N_{sn}} \mathbf{I}_{k,j}^{l,p,q} \quad (20)$$

where N_{sn} is the number of sensor nodes. $\hat{\mathbf{y}}_{k|k-1}^{l,p}$ and $\mathbf{Y}_{k|k-1}^{l,p}$ can be obtained from the results of Equations (3) and (4). The information state contribution $\mathbf{i}_{k,j}^{l,p,q}$ and the information matrix contribution $\mathbf{I}_{k,j}^{l,p,q}$ of the j th sensor are given by [4,21]

$$\mathbf{i}_{k,j}^{l,p,q} = \left(\mathbf{P}_{k|k-1}^{l,p} \right)^{-1} \mathbf{P}_{k|k-1,xz_j}^{l,p} \left(\mathbf{R}_{k,j}^q \right)^{-1} \left[\left(\mathbf{y}_{k,j} - \mathbf{z}_{k,j}^{l,p,q} \right) + \left(\mathbf{P}_{k|k-1,xz_j}^{l,p} \right)^T \left(\mathbf{P}_{k|k-1}^{l,p} \right)^{-T} \hat{\mathbf{x}}_{k|k-1}^{l,p} \right] \quad (21)$$

$$\mathbf{I}_{k,j}^{l,p,q} = \left(\mathbf{P}_{k|k-1}^{l,p} \right)^{-1} \mathbf{P}_{k|k-1,xz_j}^{l,p} \left(\mathbf{R}_{k,j}^q \right)^{-1} \left(\mathbf{P}_{k|k-1,xz_j}^{l,p} \right)^T \left(\mathbf{P}_{k|k-1}^{l,p} \right)^{-T} \quad (22)$$

Note that $\mathbf{z}_{k,j}^{l,p,q}$ and $\mathbf{P}_{k|k-1,xz_j}^{l,p}$ can be calculated by the cubature rules Equations (15) and (16), respectively, given in Section 2.1.2.

Remark 2: From Equations (19) and (20), it can be seen that the local information contributions of $\mathbf{i}_{k,j}^{l,p,q}$ and $\mathbf{I}_{k,j}^{l,p,q}$ are only computed at sensor j and the total information contribution is simply the sum of the local contributions. Therefore, the information filter is more convenient for multiple sensor estimation than the original Kalman filter.

The CCGMF needs to know the information from all sensor nodes and thus demands a large amount of communication energy, which is prohibitive for large-scale sensor networks. In the next section, an iterative diffusion-based distributed cubature Gaussian mixture filter is proposed to provide more efficient multisensor estimation.

3. Iterative Diffusion-Based Distributed Cubature Gaussian Mixture Filter

The distributed estimation lets each sensor node process its local estimation and then fuse the information from its neighboring nodes by distributed estimation algorithms to establish the global estimate. In this paper, a new distributed cubature Gaussian mixture filter based on iterative diffusion (DCGMF-ID) is introduced.

The diffusion strategy is more feasible in practice when the measurement needs to be processed in a timely manner without many iterations as in the consensus algorithm. The ordinary diffusion Kalman filter (DKF) [12–15] was designed for linear estimation problems. In this paper, the new DCGMF-ID integrates the cubature rule as well as the GM into the DKF framework to address the nonlinear distributed estimation problem. The prediction step of the DCGMF-ID at each sensor node uses the cubature rule given in Section 2.1.1. The update steps of the DCGMF-ID include the incremental update and the diffusion update, which are described as follows.

3.1. Incremental Update

Each node broadcasts its prediction information to its immediate neighbors and receives the prediction information from its immediate neighbors at the same time step. For every node j , once receiving the information, the information state and the information matrix are updated by

$$\hat{\mathbf{y}}_{k|k,j}^{l,p,q} = \hat{\mathbf{y}}_{k|k-1,j}^{l,p} + \sum_{j' \in N_j} \mathbf{i}_{k,j'}^{l,p,q} \quad (23)$$

$$\mathbf{Y}_{k|k,j}^{l,p,q} = \mathbf{Y}_{k|k-1,j}^{l,p} + \sum_{j' \in N_j} \mathbf{I}_{k,j'}^{l,p,q} \quad (24)$$

where N_j denotes the set of sensor nodes containing node j and its immediate neighbors.

3.2. Diffusion Update

As mentioned in Section 2.1.2, the number of Gaussian components after the GM reduction at each node is specified a priori to be the same, for the convenience of implementing the diffusion update. The covariance intersection algorithm can be utilized for the diffusion update. The covariance for node j can be updated by

$$\left(\mathbf{P}_{k,j}^{l,p,q}\right)^{-1} = \sum_{j' \in N_j} w_{j,j'}^{l,p,q} \left(\mathbf{P}_{k|k,j'}^{l,p,q}\right)^{-1} \quad (25)$$

or in the information matrix form

$$\mathbf{Y}_{k,j}^{l,p,q} = \sum_{j' \in N_j} w_{j,j'}^{l,p,q} \mathbf{Y}_{k|k,j'}^{l,p,q} \quad (26)$$

where $\mathbf{P}_{k|k,j'}^{l,p,q}$ denotes the covariance of the j' th sensor associated with the l, p, q th Gaussian component. $w_{j,j'}^{l,p,q}$ is the covariance intersection weight.

The state estimation for node j can be updated by

$$\left(\mathbf{P}_{k,j}^{l,p,q}\right)^{-1} \hat{\mathbf{x}}_{k,j}^{l,p,q} = \sum_{j' \in N_j} w_{j,j'}^{l,p,q} \left(\mathbf{P}_{k|k,j'}^{l,p,q}\right)^{-1} \hat{\mathbf{x}}_{k|k,j'}^{l,p,q} \quad (27)$$

or in the information state form

$$\hat{\mathbf{y}}_{k,j}^{l,p,q} = \sum_{j' \in N_j} w_{j,j'}^{l,p,q} \hat{\mathbf{y}}_{k|k,j'}^{l,p,q} \quad (28)$$

The weights $w_{j,j'}^{l,p,q}$ are calculated by [22]

$$\begin{cases} w_{j,j'}^{l,p,q} = \frac{1/\text{tr}\left(\left(\mathbf{Y}_{k|k,j'}^{l,p,q}\right)^{-1}\right)}{\sum_{j' \in N_j} 1/\text{tr}\left(\left(\mathbf{Y}_{k|k,j'}^{l,p,q}\right)^{-1}\right)}, & j' \in N_j \\ w_{j,j'}^{l,p,q} = 0, & j' \notin N_j \end{cases} \quad (29)$$

where $\text{tr}(\cdot)$ denotes the trace operation.

Remark 3: Different from the conventional diffusion-based distributed estimation algorithms, the DCGMF-ID performs the diffusion update multiple times iteratively, rather than updating it only once. The advantage of the iterative diffusion update is that estimates from different sensors eventually converge.

The DCGMF-ID algorithm (Algorithm 1) can be summarized as follows:

Algorithm 1

- Step 1:** Each sensor node calculates the local prediction using Equations (3) and (4), and the cubature rule, and transforms them to the information state $\hat{\mathbf{y}}_{k|k-1,j}^{l,p}$ and the information matrix $\mathbf{Y}_{k|k-1,j}^{l,p}$.
- Step 2:** When new measurements are available, each node evaluates the information state contribution $\hat{\mathbf{y}}_{k,j}^{l,p,q}$ and the information matrix contribution $\mathbf{Y}_{k|k,j}^{l,p,q}$ by using Equations (21) and (22).
- Step 3:** Each node communicates with its immediate neighbors to update its information state and information matrix through the incremental update (i.e., Equations (23) and (24)).
- Step 4:** Each node runs the diffusion update by Equations (26) and (28) multiple times. Let t denote the t th iteration of the diffusion update. The iterative diffusion updates can be given by

$$\hat{\mathbf{y}}_{k|k,j}^{l,p,q}(t+1) = \sum_{j' \in N_j} w_{j,j'}^{l,p,q}(t) \hat{\mathbf{y}}_{k|k,j'}^{l,p,q}(t) \quad (30a)$$

$$\mathbf{Y}_{k|k,j}^{l,p,q}(t+1) = \sum_{j' \in N_j} w_{j,j'}^{l,p,q}(t) \mathbf{Y}_{k|k,j'}^{l,p,q}(t) \quad (30b)$$

When $t = t_{\max}$, the final estimates are $\mathbf{Y}_{k,j}^{l,p,q} = \mathbf{Y}_{k|k,j}^{l,p,q}(t_{\max})$; $\hat{\mathbf{y}}_{k,j}^{l,p,q} = \hat{\mathbf{y}}_{k|k,j}^{l,p,q}(t_{\max})$

Calculate the mean $\hat{\mathbf{x}}_{k|k}^{l,p,q}$ and covariance $\mathbf{P}_{k|k}^{l,p,q}$ of each Gaussian component.

The final GM can be represented by $\sum_{l=1}^{N_l} \sum_{p=1}^{N_p} \sum_{q=1}^{N_q} \alpha^l \alpha^p \alpha^q N(\mathbf{x}; \hat{\mathbf{x}}_{k|k}^{l,p,q}, \mathbf{P}_{k|k}^{l,p,q})$.

Step 5: Conduct GM reduction.

Step 6: Let $k = k + 1$; continues to Step 1.

The iterative diffusion update is identical to the iterative covariance intersection (ICI) algorithm [16]. Thus, the proposed distributed estimation has the same properties of unbiasedness and consistency as the ICI. For linear systems, if the initial estimate at each sensor node is unbiased, the estimate through the incremental update and the diffusion update in each filtering cycle is still unbiased. For nonlinear systems, however, the unbiasedness may not be preserved. It is also true for the analysis of consistency. When the covariance intersection (CI) method is used for data fusion, consistency is ensured based on the assumption that the estimate at each sensor node is consistent [23]. If it is assumed that each node's local estimate after the incremental step is consistent

(i.e., $P_{k|k,j} \geq E \left[\left(\hat{x}_{k|k,j} - x_k \right) \left(\hat{x}_{k|k,j} - x_k \right)^T \right]$), then by the diffusion update, the fused estimate is still consistent because the CI is applied. Without this assumption, consistency is not guaranteed by the CI technique. For linear systems, this assumption can be easily met and consistency can be guaranteed. For nonlinear systems, the high-degree (fifth-degree) cubature rule based-filtering is utilized in this paper for the local estimation at each node. It can provide more accurate estimate of $\hat{x}_{k|k,j}$ and $P_{k|k,j}$ than the third-degree cubature Kalman filter (CKF) and the unscented Kalman filter (UKF). Therefore, although the unbiasedness and consistency cannot be guaranteed for nonlinear systems, they can be better approached by the proposed distributed estimation scheme than other distributed nonlinear filters.

It is necessary to compare the DCGMF-ID with the consensus-based distributed estimation. For the iterative diffusion strategy in the DCGMF-ID, if the local estimate obtained at each node after the incremental update is consistent, the fused estimate by the diffusion update is also consistent, no matter how many iterations of the iterative diffusion update since the CI is applied. In addition, it was shown in [16] that the covariance and estimate from each node converge to a common value (i.e., $\lim_{t \rightarrow \infty} P_{k,j}(t) = P_k$ and $\lim_{t \rightarrow \infty} \hat{x}_{k,j}(t) = \hat{x}_k$). Recall that “ t ” represents the t th diffusion iteration, not the time. However, for the consensus-based distributed estimation [24], even if the local estimate obtained at each node is consistent, if the number of iterations of consensus is not infinite, the consistency of the fused estimate cannot be preserved [24]. Because the average consensus cannot be achieved in a few iterations, a multiplication by $|N|$, the cardinality of the network, will lead to an overestimate of the information, which is not desirable. Although another approach was proposed in [24]—to fuse the information from each node in order to preserve consistency—the new consensus algorithm results in more computation complexity.

In the following, we provide a more complete analysis of the convergence by the following two propositions.

Proposition 1: *The iterative diffusion update Equations (30a) and (30b) can be represented in a general form of $\eta(t+1) = A(t)\eta(t)$, where each (j, j') entry of the transition matrix $A(t)$ denoted by $a_{j,j'}(t)$ corresponds to the weight $w_{j,j'}^{l,p,q}(t)$. Assume that the sensor network is connected. If there exists a positive constant $\alpha < 1$ and the following three conditions are satisfied*

- (a) $a_{j,j}(t) \geq \alpha$ for all j, t ;
- (b) $a_{j,j'}(t) \in \{0\} \cup [\alpha, 1]$, $j \neq j'$;
- (c) $\sum_{j'=1}^{N_{sn}} a_{j,j'}(t) = 1$ for all j, j', t ;

the estimates using the proposed DCGMF-ID reach a consensus value.

Proof: The proof uses the theorem 2.4 in [25]. If the connected sensor network satisfies these three conditions, $\eta(t)$, using the algorithm:

$$\eta(t+1) = A(t)\eta(t) \quad (31)$$

converges to a consensus value. For the scalar case (the dimension of the state is one), $a_{j,j'}(t)$ corresponds to $w_{j,j'}^{l,p,q}(t)$. The j th element of $\eta(t)$ corresponds to the information state $\hat{y}_{k|k,j}^{l,p,q}(t)$. For the vector case, the transition matrix $A(t) \otimes I_n$ should be applied where \otimes denotes the Kronecker product and n is the dimension of the state. For the matrix case, each column of the matrix can be treated as the vector case. \square

As seen from Equation (29), the weight $w_{j,j'}^{l,p,q}(t)$ only depends on the covariance matrix. Here we assume that the covariance in the first iteration is upper bounded, and for any t there is no covariance matrix equal to 0 (no uncertainty). As long as node j and node j' are connected, $w_{j,j'}^{l,p,q}(t) \in (0, 1)$. Thus,

condition (b) is satisfied. In addition, from Equation (29), $\sum_{j'=1}^{N_{sn}} w_{j,j'}^{l,p,q}(t) = 1$ always holds; that is, the transition matrix $A(t)$ is always row-stochastic. Therefore, condition (c) is satisfied.

For any arbitrary large t , say t_{max} , the non-zero weight set $\{w_{j,j'}^{l,p,q}(t), t = 1, \dots, t_{max}\}$ for all j, j' is a finite set since the number of nodes and the number of iterations are finite. There always exists a minimum value in this finite set. Thus, α can be chosen to be $0 < \alpha \leq \min\{w_{j,j'}^{l,p,q}(t)\}$ such that conditions (a) and (b) are satisfied.

According to the theorem 2.4 in [25] for the agreement algorithm Equation (31), the estimate $\eta(t)$ reaches a consensus value.

Proposition 2: *If the assumption and conditions in Proposition 1 are satisfied, the consensus estimate using the DCGMF-ID is unique.*

Proof: Let $U_{0,t} = A(t)A(t-1) \cdots A(0)$ be the backward product of the transition matrices and $\lim_{t \rightarrow \infty} U_{0,t} = U^*$ according to Proposition 1. On the other hand, when the consensus is achieved, the covariance matrix or the information matrix $Y_{k|k,j'}^{l,p,q}$ associated with each node becomes the same. According to Equation (29), the weights $w_{j,j'}^{l,p,q}(t)$ converge to the same value. Thus, $\lim_{t \rightarrow \infty} A(t) = A^*$ and $A^* \mathbf{1} = \mathbf{1}$ since A^* is a row-stochastic matrix where $A^* = [a_1 \ a_2 \ \cdots \ a_n]^T$ with a_j being the row vector of the matrix A^* . Furthermore, because $Y_{k|k,j'}^{l,p,q}$ converges to the same value, from Equation (29), all the non-zero weights $w_{j,j'}^{l,p,q}(t)$ or all non-zero entries of the row vector a_j are identical and equal to the reciprocal of the degree of the j th node, i.e., $\frac{1}{\delta_j}$ (where $\delta_j \triangleq$ degree of the j th node \triangleq cardinality of N_j). Hence, A^* is deterministic given the connected sensor network. \square

A^* is irreducible since the sensor network is connected. Moreover, the diagonal elements of A^* are all positive (equal to the reciprocal of the degree of each node). Hence, 1 is a unique maximum eigenvalue of A^* [26] and, in fact, A^* is a primitive matrix [26].

In the sense of consensus, $\lim_{t \rightarrow \infty} \eta(t) = U^* \eta(0)$, we have $A^* U^* = U^*$ or $(A^* - I)U^* = \mathbf{0}$ (note, it is not possible for U^* to be $\mathbf{0}$ since it is the backward product of non-negative matrices). The column of U^* belongs to the null space of $A^* - I$. Since 1 is the unique maximum eigenvalue of A^* , 0 is the unique eigenvalue of $A^* - I$ and the dimension of the null space of $A^* - I$ is 1. Thus, $\mathbf{1}$ (or any scalar multiplication of $\mathbf{1}$) is the unique vector belonging to the null space of $A^* - I$. Therefore, U^* is ergodic, i.e., $U^* = \mathbf{1} [\alpha_1, \alpha_2, \dots, \alpha_n]$ where α_i is a scalar constant. According to Theorem 4.20 in [27], $[\alpha_1, \alpha_2, \dots, \alpha_n]$ and the consensus value of $\eta(t)$ are unique.

The iterative diffusion update in the DCGMF-ID can be interpreted from the information theory perspective as the process of minimizing the Kullback–Leibler divergence (KLD) [28]. In the information theory, a measure of distance between different pdfs can be given by the KLD. Given the local pdf p_i with the weight π_i , the fused pdf p_f can be obtained by minimizing the KLD:

$$p_f = \operatorname{argmin}_p \sum_{i=1}^{N_{sn}} \pi_i D(p||p_i) \tag{32}$$

with $\sum_{i=1}^{N_{sn}} \pi_i = 1$ and $\pi_i \geq 0$. $D(p||p_i)$ is the KLD defined as:

$$D(p||p_i) = \int p(x) \log \frac{p(x)}{p_i(x)} dx \tag{33}$$

The KLD is always non-negative, and equal to zero only when $p(x) = p_i(x)$. The solution to Equation (32) turns out to be [28]

$$p_f(x) = \frac{\prod_{i=1}^{N_{sn}} [p_i(x)]^{\pi_i}}{\int \prod_{i=1}^{N_{sn}} [p_i(x)]^{\pi_i} dx} \tag{34}$$

The above equation is also the Chernoff fusion [29]. Under the Gaussian assumption, which is true for each component of the GM model in this paper, it was shown in [29] that the Chernoff fusion yields update equations identical to the covariance intersection Equations (25)–(28).

Therefore, from the information-theoretic perspective, the iterative diffusion update Equation (30) is actually equivalent to minimizing the KLD repeatedly. For instance, the diffusion update at the t th iteration is equivalent to

$$p_{f,j}(t+1) = \arg \min_{p_j(t+1)_{j' \in N_j}} \sum \omega_{j,j'}(t) D(p_j(t+1) \parallel p_{j'}(t)) \text{ with } j = 1, \dots, N_{sn} \tag{35}$$

When t approaches t_{max} , from the convergence property of the iterative diffusion (i.e., Propositions 1 and 2), the cost for the minimization problem in Equation (35) approaches 0 since $p_j(t_{max}) = \bar{p}$ for all $j = 1, \dots, N_{sn}$, and $D(\bar{p} \parallel \bar{p}) = 0$ where \bar{p} is the final convergent pdf.

4. Numerical Results and Analysis

In this section, the performance of DCGMF based on different fusion strategies is demonstrated via a benchmark target-tracking problem using multiple sensors, which is to track a target executing a maneuvering turn in a two-dimensional space with unknown and time-varying turn rate [3]. The target dynamics is highly nonlinear due to the unknown turn rate. It has been used as a benchmark problem to test the performance of different nonlinear filters [3,30].

The discrete-time dynamic equation of the target motion is given by:

$$\mathbf{x}_k = \begin{bmatrix} 1 & \frac{\sin(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 0 & \frac{\cos(\omega_{k-1}\Delta t)-1}{\omega_{k-1}} & 0 \\ 0 & \cos(\omega_{k-1}\Delta t) & 0 & -\sin(\omega_{k-1}\Delta t) & 0 \\ 0 & \frac{1-\cos(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 1 & \frac{\sin(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 0 \\ 0 & \sin(\omega_{k-1}\Delta t) & 0 & \cos(\omega_{k-1}\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{36}$$

where $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, \omega_k]^T$; $[x_k, y_k]$ and $[\dot{x}_k, \dot{y}_k]$ are the position and velocity at time k , respectively; Δt is the time-interval between two consecutive measurements; ω_{k-1} is the unknown turn rate at the time $k - 1$; and \mathbf{v}_{k-1} is the white Gaussian noise with mean zero and covariance \mathbf{Q}_{k-1} ,

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 1.75 \times 10^{-4} \Delta t \end{bmatrix} \tag{37}$$

The measurements are the range and angle given by

$$\mathbf{y}_k = \begin{pmatrix} \sqrt{x_k^2 + y_k^2} \\ \text{atan2}(y_k, x_k) \end{pmatrix} + \mathbf{n}_k \tag{38}$$

where atan2 is the four-quadrant inverse tangent function; \mathbf{n}_k is the measurement noise with an assumed non-Gaussian distribution $\mathbf{n}_k \sim 0.5N(\bar{\mathbf{n}}_1, \mathbf{R}_1) + 0.5N(\bar{\mathbf{n}}_2, \mathbf{R}_2)$, where $\bar{\mathbf{n}}_1 = [5 \text{ m}, -2 \times 10^{-6} \text{ mrad}]^T$ and $\bar{\mathbf{n}}_2 = [-5 \text{ m}, 0 \text{ mrad}]^T$. The variances \mathbf{R}_1 and \mathbf{R}_2 are assumed

to be $\mathbf{R}_1 = \text{diag} \left(\left[100 \text{ m}^2, 10 \text{ mrad}^2 \right] \right)$ and $\mathbf{R}_2 = \begin{bmatrix} 80 \text{ m}^2 & 10^{-1} \text{ mrad} \\ 10^{-1} \text{ mrad} & 10 \text{ mrad}^2 \end{bmatrix}$. The sampling interval is $\Delta t = 1 \text{ s}$. The simulation results are based on 100 Monte Carlo runs. The initial estimate $\hat{\mathbf{x}}_0$ is generated randomly from the normal distribution $N(\hat{\mathbf{x}}_0; \mathbf{x}_0, \mathbf{P}_0)$ with \mathbf{x}_0 being the true initial state $\mathbf{x}_0 = [1000 \text{ m}, 300 \text{ m/s}, 1000 \text{ m}, 0, -3 \text{ deg/s}]^T$ and \mathbf{P}_0 being the initial covariance $\mathbf{P}_0 = \text{diag} \left(\left[100 \text{ m}^2, 10 \text{ m}^2/\text{s}^2, 100 \text{ m}^2, 10 \text{ m}^2/\text{s}^2, 100 \text{ mrad}^2/\text{s}^2 \right] \right)$. Sixteen sensors are used in simulation. The topology of the sensor network is shown in Figure 2. Note that the “circle” denotes the sensor node. It is assumed that the target is always in the range and field of view of all sensors.

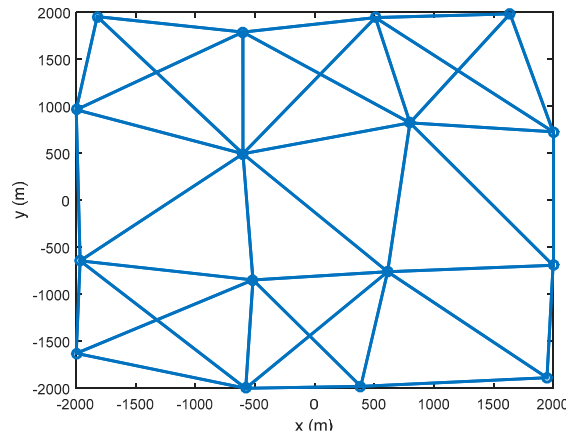


Figure 2. The network of sensors.

The metric used to compare the performance of different filters is the root mean square error (RMSE). The RMSEs of the position, velocity, and turn rate using different filters with the third-degree cubature rule are shown in Figures 3–5, respectively. The cubature Gaussian mixture filter (CGMF) using a single sensor, the distributed cubature Gaussian mixture filter based on the iterative covariance intersection [16] (DCGMF-ICI), average consensus (DCGMF-AC), iterative diffusion strategies (DCGMF-ID), and the centralized cubature Gaussian mixture filter (CCGMF) are compared. Since DCGMF-ICI, DCGMF-AC, and DCGMF-ID all involve iterations, it is more illustrative to use the number of iterations as a parameter to compare their performance. “M” in the figures is the iteration number.

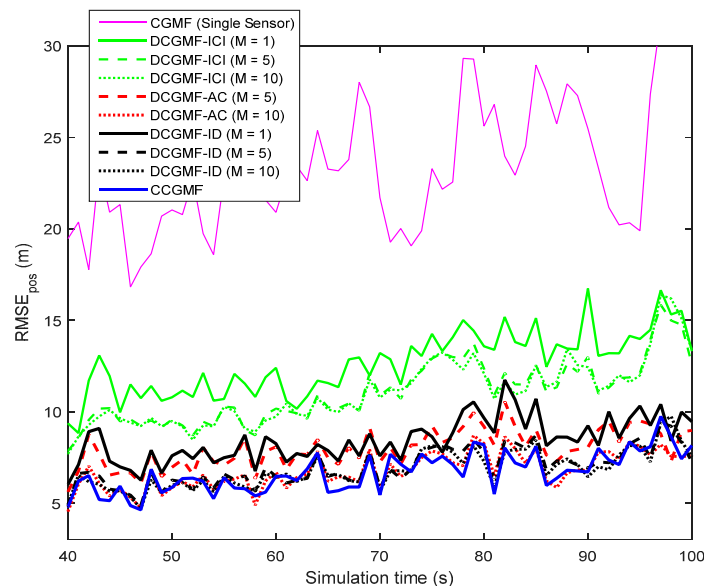


Figure 3. Root mean square errors (RMSEs) of the position estimation.

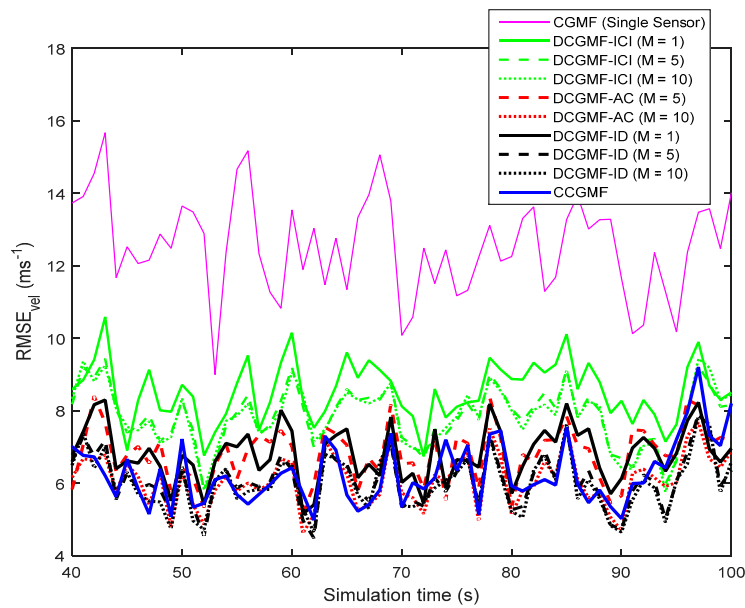


Figure 4. RMSEs of the velocity estimation.

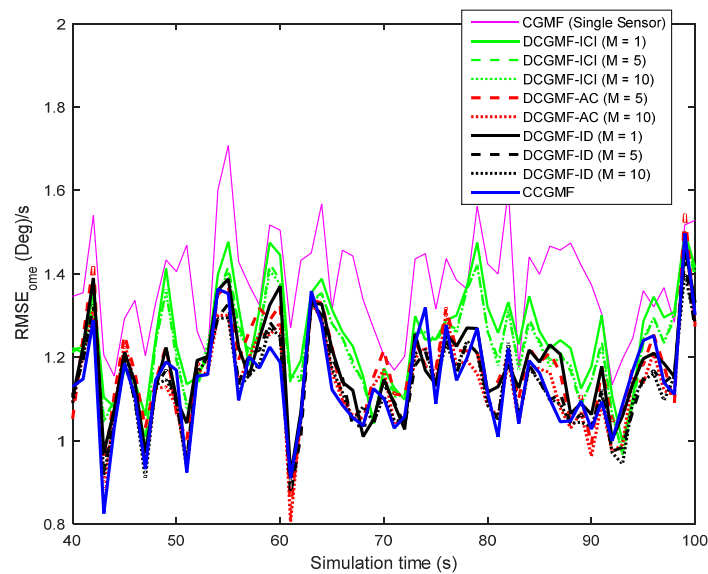


Figure 5. RMSEs of the turn-rate estimation.

It can be seen from the figures that (1) DCGMFs and CCGMF exhibit better performance than CGMF using single sensor since more information from multiple sensors can be exploited; (2) with the increase of iterations, the performance of all DCGMFs is improved; (3) the DCGMF-ICI is less accurate than the DCGMF-AC and the DCGMF-ID since the ICI algorithm does not do the incremental update; (4) both the DCGMF-AC ($M = 10$) and the DCGMF-ID ($M = 10$) achieve very close performance to the CCGMF. However, fewer iterations have a more negative effect on the performance of the DCGMF-AC than that on the DCGMF-ID. The DCGMF-ID is more effective in terms of iterations since the DCGMF-ID with $M = 1$ has close performance to the DCGMF-AC with $M = 5$. Hence, when the allowable number of information exchanges is limited, DCGMF-ID would be the best filter. It is also worth noting that the DCGMF-AC requires less computational effort at each node, but requires more communication expense than the DCGMF-ID. If the communication capability of the sensor network is not a main constraint, the DCGMF-AC would be a competitive approach.

Next, we compare the performance of DCGMFs using the third-degree cubature rule and the DCGMFs using the fifth-degree cubature rule. The metric is the averaged cumulative RMSE (CRMSE). The CRMSE for the position is defined by

$$\text{CRMSE}_{\text{pos}} = \frac{1}{N_{mc}} \sum_{m=1}^{N_{mc}} \sqrt{\frac{1}{N_{sim}} \sum_{i=1}^{N_{sim}} \sum_{j=1,3} \left(x_{i,m}^j - \hat{x}_{i,m}^j \right)^2} \quad (39)$$

where $N_{sim} = 100$ s is the simulation time and $N_{mc} = 100$ is the number of Monte Carlo runs. The superscript “ j ” denotes the j th state variable and the subscripts “ i ” and “ m ” denote the i th simulation time step and the m th simulation, respectively. The CRMSE for the velocity and CRMSE for the turn rate can be similarly defined.

The results of DCGMF-AC using the third-degree cubature rule and the fifth-degree cubature rule show indistinguishable difference. Similar results can be observed for CCGMF. DCGMF-ID and DCGMF-ICI using the fifth-degree cubature rule, however, show better performance than those using the third-degree cubature rule. The reason is that DCGMF-ID and DCGMF-ICI depend heavily on the local measurement to perform estimation, while DCGMF-AC and CCGMF update the estimate based on global observations from all sensors. Specifically, for DCGMF-AC, although each sensor communicates measurement only with its neighbors, after convergence of the consensus iterations, each sensor actually obtained a fused measurement information from all sensors. Because the high degree numerical rule affects the accuracy of estimates extracted from the observations, the fifth-degree cubature rule can more noticeably improve the performance of DCGMF-ID and DCGMF-ICI based on only local observations. However, the benefit of using the high-degree numerical rule will be mitigated if more information from more sensors is available as for the DCGMF-AC and CCGMF. Hence, we only compare the results of DCGMF-ID and DCGMF-ICI using the third-degree and the fifth-degree cubature rules in Table 1. In order to see merely the effect of the cubature rules with different accuracy degrees on the performance, we want to minimize the effect of different iterations on the performance of different filters. Therefore, a sufficiently large iteration number, $M = 20$, is used to ensure that the different filters already converge after iterations. It can be seen from the Table 1 that DCGMF-ID and DCGMF-ICI using the fifth-degree cubature rule can achieve better performance than those using the third-degree cubature rule.

Table 1. Cumulative root mean square errors (CRMSEs) of different filters.

Filters	CRMSE (Position)	CRMSE (Velocity)	CRMSE (Turn Rate)
DCGMF-ID (third-degree, $M = 20$)	5.85892	5.60166	0.019392
DCGMF-ID (fifth-degree, $M = 20$)	5.78748	5.57730	0.019375
DCGMF-ICI (third-degree, $M = 20$)	8.81274	7.22025	0.020837
DCGMF-ICI (fifth-degree, $M = 20$)	8.02142	7.11939	0.020804

Distributed cubature Gaussian mixture filter based on an iterative diffusion strategy (DCGMF-ID) and DCGMF based on the iterative covariance intersection (DCGMF-ICI).

5. Conclusions

A new iterative diffusion-based distributed cubature Gaussian mixture filter (DCGMF-ID) was proposed for the nonlinear non-Gaussian estimation using multiple sensors. The convergence property of the DCGMF-ID was analyzed. It has been shown via a target-tracking problem that the DCGMF-ID can successfully approximate the performance of the centralized cubature Gaussian mixture filter and has all the advantages of the distributed filters. Among the iterative distributed estimation strategies, the DCGMF-ID exhibits more accurate results than the iterative covariance intersection based method (i.e., DCGMF-ICI). It also shows better performance than the average consensus-based method given the same number of iterations. In addition, the fifth-degree cubature rule can improve the accuracy of the DCGMF-ID.

Acknowledgments: This work was supported by the US National Science Foundation (NSF) under the grant NSF-ECCS-1407735.

Author Contributions: All authors contributed significantly to the work presented in this manuscript. Bin Jia conceived the original concept, conducted the numerical simulations, and wrote the initial draft of the paper. Tao Sun and Ming Xin provided the theoretical analysis and proofs of the results. Ming Xin contributed to the detailed writing for the initial submission, the revision of the manuscript, and the response to the reviewers' comments. All authors actively participated in valuable technical discussions in the process of completing the paper.

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

References

1. Alspach, D.L.; Sorenson, H.W. Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Trans. Autom. Control* **1972**, *17*, 439–448. [[CrossRef](#)]
2. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Proc.* **2002**, *50*, 174–188. [[CrossRef](#)]
3. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [[CrossRef](#)]
4. Jia, B.; Xin, M. Multiple sensor estimation using a new fifth-degree cubature information filter. *Trans. Inst. Meas. Control* **2015**, *37*, 15–24. [[CrossRef](#)]
5. Gelb, A. *Applied Optimal Estimation*; MIT Press: Cambridge, MA, USA, 1974; pp. 182–202.
6. Juier, S.J.; Uhlmann, J.K.; Durrant-Whyte, H.F. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **2000**, *45*, 477–482.
7. Olfati-Saber, R. Distributed Kalman filter with embedded consensus filters. In Proceedings of the 44th IEEE Conference on Decision and Control, and 2005 European Control Conference, Seville, Spain, 15 December 2005; pp. 8179–8184.
8. Olfati-Saber, R. Distributed Kalman filter for sensor networks. In Proceedings of the 46th IEEE on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.
9. Kamal, A.T.; Farrell, J.A.; Roy-Chodhury, A.K. Information weighted consensus. In Proceedings of the 51st IEEE on Decision and Control, Maui, HI, USA, 10–13 December 2012; pp. 2732–2737.
10. Kamal, A.T.; Farrell, J.A.; Roy-Chodhury, A.K. Information weighted consensus filters and their application in distributed camera networks. *IEEE Trans. Autom. Control* **2013**, *58*, 3112–3125. [[CrossRef](#)]
11. Sun, T.; Xin, M.; Jia, B. Distributed estimation in general directed sensor networks based on batch covariance intersection. In Proceedings of the 2016 American Control Conference, Boston, MA, USA, 6–8 July 2016; pp. 5492–5497.
12. Cattivelli, F.S.; Sayed, A.H. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Trans. Autom. Control* **2010**, *55*, 2069–2084. [[CrossRef](#)]
13. Cattivelli, F.S.; Lopes, C.G.; Sayed, A.H. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. Autom. Control* **2008**, *56*, 1865–1877. [[CrossRef](#)]
14. Cattivelli, F.S.; Sayed, A.H. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Autom. Control* **2010**, *55*, 1035–1048. [[CrossRef](#)]
15. Hu, J.; Xie, L.; Zhang, C. Diffusion Kalman filtering based on covariance intersection. *IEEE Trans. Signal Proc.* **2012**, *60*, 891–902. [[CrossRef](#)]
16. Hlinka, O.; Sluciak, O.; Hlawatsch, F.; Rupp, M. Distributed data fusion using iterative covariance intersection. In Proceedings of the 2014 IEEE on Acoustics, Speech, and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 1880–1884.
17. Li, W.; Jia, Y. Distributed consensus filtering for discrete-time nonlinear systems with non-Gaussian noise. *Signal Proc.* **2012**, *92*, 2464–2470. [[CrossRef](#)]
18. Runnalls, A.R. Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 989–999. [[CrossRef](#)]
19. Salmond, D. Mixture reduction algorithms for target tracking in clutter. In Proceedings of the SPIE 1305 Signal and Data Processing of Small Targets, Los Angeles, CA, USA, 16–18 April 1990; pp. 434–445.
20. Williams, J.L. Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter. Master's Thesis, Air Force Institute of Technology, Dayton, OH, USA, 2003.

21. Jia, B.; Xin, M.; Cheng, Y. Multiple sensor estimation using the sparse Gauss-Hermite quadrature information filter. In Proceedings of the 2012 American Control Conference, Montreal, QC, Canada, 27–29 June 2012; pp. 5544–5549.
22. Niehsen, W. Information fusion based on fast covariance intersection. In Proceedings of the 2002 5th International Conference on Information Fusion, Annapolis, MD, USA, 8–11 July 2002.
23. Julier, S.; Uhlmann, J. General decentralized data fusion with covariance intersection (CI). In *Handbook of Multisensor Data Fusion*; CRC Press: Boca Raton, FL, USA, 2009.
24. Battistelli, G.; Chisci, L. Consensus-based linear and nonlinear filtering. *IEEE Trans. Autom. Control* **2015**, *60*, 1410–1415. [[CrossRef](#)]
25. Alex, O.; Tsitsiklis, J.N. Convergence speed in distributed consensus and averaging. *SIAM Rev.* **2011**, *53*, 747–772.
26. Horn, R.; Johnson, C. *Matrix Analysis*; Cambridge University Press: New York, NY, USA, 1985.
27. Seneta, E. *Nonnegative Matrices and Markov Chains*; Springer: New York, NY, USA, 2006.
28. Battistelli, G.; Chisci, L. Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica* **2014**, *50*, 707–718. [[CrossRef](#)]
29. Hurley, M.B. An information theoretic justification for covariance intersection and its generalization. In Proceedings of the 5th International Conference on Information Fusion, Annapolis, MD, USA, 8–11 July 2002.
30. Jia, B.; Xin, M.; Cheng, Y. High degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [[CrossRef](#)]



© 2016 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/>).