Signal Processing 92 (2012) 2464–2470

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Distributed consensus filtering for discrete-time nonlinear systems with non-Gaussian noise

Wenling Li [a,b,*], Yingmin Jia [a,b,c]

[a] The Seventh Research Division, Beihang University (BUAA), Beijing 100191, China
[b] The Department of Systems and Control, Beihang University (BUAA), Beijing 100191, China
[c] Key Laboratory of Mathematics, Informatics and Behavioral Semantics (LMIB), Ministry of Education, SMSS, Beihang University (BUAA), Beijing 100191, China

ARTICLE INFO

Article history:
Received 30 November 2010
Received in revised form
2 January 2012
Accepted 11 March 2012
Available online 28 March 2012

Keywords:
Distributed estimation
Gaussian mixture
Unscented Kalman filter
Average-consensus

ABSTRACT

This paper studies the problem of distributed estimation for a class of discrete-time nonlinear non-Gaussian systems in a not fully connected sensor network environment. The non-Gaussian process noise and measurement noise are approximated by finite Gaussian mixture models. A distributed Gaussian mixture unscented Kalman filter (UKF) is developed in which each sensor node independently calculates local statistics by using its own measurement and an average-consensus filter is utilized to diffuse local statistics to its neighbors. A main difficulty encountered is the distributed computation of the Gaussian mixture weights, which is overcome by introducing the natural logarithm transformation. The effectiveness of the proposed distributed filter is verified via a simulation example involving tracking a target in the presence of glint noise.

1. Introduction

Distributed estimation has been one of the most fundamental collaborative information processing problems in large scale sensor networks [1–3]. This is partly due to its broad applications in robotics, surveillance and target tracking. Unlike the conventional centralized fusion solution, which requires all the sensor nodes send their measurements to a central processing unit and may need a large amount of energy for communications, the distributed strategy requires fewer communications and allows parallel processing. On the other hand, compared with the fully decentralized estimation, the sensor network is not needed to be fully connected and each sensor node only communicates with its neighboring peers in the distributed framework.

As the sensor network is not fully connected, the distributed estimation is implemented based on the idea that each sensor node independently calculates the local sufficient statistics by using its own measurement, and an average-consensus filter is employed to diffuse the local statistics over the entire sensor network through communicating with neighboring peers so that the global statistics can be derived by each sensor node. It can be observed that the distributed algorithm is scalable since each sensor node only requires local communication and it is also robust to node failures given that the sensor network is still connected. Many approaches have been proposed to resolve the distributed estimation, of which the Kalman filter has been one of the most significant candidates. For instance, the distributed Kalman filter has been proposed for discrete-time linear Gaussian systems by Olfati-Saber [4–6]. Although these algorithms are formulated for linear systems, it is straightforward to adapt them for use with nonlinear systems by employing the extended Kalman filter (EKF). However, the EKF performs poorly in certain situations such as the nonlinearity is severe. The unscented

* Corresponding author at: The Seventh Research Division, Beihang University (BUAA), Beijing 100191, China. Tel.: +86 10 8233 8683; fax: +86 10 8231 6100.
E-mail addresses: lwlmath@gmail.com (W. Li), ymjia@buaa.edu.cn (Y. Jia).

0165-1684/$ - see front matter
http://dx.doi.org/10.1016/j.sigpro.2012.03.009

Kalman filter (UKF), which uses the unscented transform to approximate the moments required in the exact Kalman filter, has been shown to be more accurate with the same computational cost [7]. The distributed version of UKF has been developed in [8]. Note that the aforementioned work focus on the systems with Gaussian noise, which might not hold in many practical applications.

For the state estimation of discrete-time nonlinear non-Gaussian systems, the sequential Monte Carlo method or the particle filter has been considered as an effective solution [9]. Distributed particle filters (DPFs) have been developed by using different strategies [10–15]. In [10,11], using an average-consensus algorithm, DPFs are derived by broadcasting local statistics of weighted particles between sensors. In [12], an alternative approach is developed which requires only exchange particle weights among sensors. The likelihood consensus scheme is employed in [13,14] to approximate the joint likelihood function in a distributed way. An asynchronous DPF is presented in [15] by using a gossiping protocol. Another alternative for nonlinear non-Gaussian systems is to approximate the non-Gaussian probability density function by Gaussian mixture models [16]. Specially, the authors in [16] presented a detailed discussion on how a posterior density can be closely approximated by a weighted sum of Gaussian probability density functions. It should be pointed out that the nonlinearity of models might lead to a non-Gaussian multi-modal probability density function even when the system and the measurement noise processes are Gaussian and the Gaussian approximation of this multi-modal distribution will result in degraded performance. Much effort has been devoted to the problem of filtering in Gaussian mixture noise environment [17–20].

In this paper, we attempt to propose a distributed Gaussian mixture UKF (GM-UKF) for a class of discrete-time nonlinear systems with non-Gaussian noise. By using the Gaussian mixture models to approximate the non-Gaussian noise, the distributed UKF is extended to develop a distributed GM-UKF. The main difficulty encountered is that the distributed computation of the mixture weights involves a product term of likelihood functions and therefore the average-consensus filter cannot be applied directly. This difficulty is overcome by employing natural logarithm transformation. As the number of Gaussian components increases exponentially as time progresses, a simple pruning scheme is utilized to keep this number constant at each time step. Simulation results are provided to illustrate the effectiveness of the proposed filter.

The remainder of this paper is organized as follows. The problem of distributed estimation is formulated in Section 2. In Section 3, distributed GM-UKF for nonlinear systems with non-Gaussian noise is developed by the average-consensus theory. In Section 4, a numerical example is provided to illustrate the effectiveness of the proposed filter. Conclusion is drawn in Section 5.

2. Problem formulation

Consider the following discrete-time nonlinear system:

$$x_k = f(x_{k-1}) + w_{k-1} \tag{1}$$

$$z_k^i = h^i(x_k) + v_k^i, \quad i = 1,2,\ldots,N \tag{2}$$

where $x_k \in \mathbb{R}^n$ and $z_k^i \in \mathbb{R}^m$ are the system state and the ith sensor measurement vectors, respectively. N is the number of sensor nodes. f and h^i are the system transition and the ith sensor measurement functions, respectively. The process noise w_{k-1} and the measurement noise v_k^i are non-Gaussian processes. The communication topology between sensor nodes is modeled as a undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ where $\mathcal{V} = \{1,2,\ldots,N\}$ denotes the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes the edge set. The nodes with which a node communicates are called its neighbors. It is assumed that each sensor node can only communicate with its neighbors but not all the other sensor nodes. In addition, the graph $\mathcal{G}$ is assumed to be connected (i.e., sensor nodes may not be able to communicate directly but there is path of that allows information to travel from any node to any other node) to preserve the averaging in the following average-consensus filters. It is worth mentioning here that many strategies have been proposed to reach consensus for different types of graphs [21–23].

In the real world applications, the non-Gaussian noise is more difficult to handle than Gaussian noise and the Gaussian mixture models have been suggested as approximations to non-Gaussian densities. The main idea of the Gaussian mixture models is that any probability density function can be closely approximated by a sum of a finite number of weighted Gaussian terms. In this work, we endeavor to provide a general framework for distributed filtering by handling non-Gaussian noise processes as Gaussian mixture models, i.e., the probability density functions of w_k and v_k^i can be approximated by

$$p(w_k) = \sum_{p=1}^{G_1} \alpha_p \mathcal{N}(w_k; \mu_{k,p}, Q_{k,p}) \tag{3}$$

$$p(v_k^i) = \sum_{q=1}^{G_2} \beta_q \mathcal{N}(v_k^i; v_{k,q}^i, R_{k,q}^i) \tag{4}$$

where α_p and β_q are the mixture weights. G_1 and G_2 are the numbers of mixture components. The mixture weights are required to sum up to one. $\mathcal{N}(x;\sigma,\Sigma)$ represents the probability density function of a Gaussian distributed random variable x with mean σ and covariance Σ. Without loss of generality, we assume Gaussian terms are with zero-mean, i.e., $\mu_{k,p} = v_{k,q}^i = 0$.

The purpose of this paper is to develop a distributed filtering algorithm for the discrete-time nonlinear system (1) and (2) with Gaussian mixture noise processes (3) and (4). As each sensor node can only communicate with its neighbors but not all the other sensor nodes, the performance of the distributed filter is expected to achieve that of the centralized fusion filter which involves all the sensor measurements.

3. Distributed Gaussian mixture unscented Kalman filter

To address the problem of distributed estimation for discrete-time nonlinear system (1) and (2) with non-Gaussian noise (3) and (4), we first derive the recursive formula as the conventional centralized Kalman filter.

More precisely, we assume that each sensor node can communicate with all the other nodes and therefore each sensor node can obtain all the sensor measurements. On this occasion, each sensor node can be considered as a fusion center. To be specific, assume that the posterior distribution can be represented as a Gaussian mixture with L terms

$$p(x_{k-1}|Z_{k-1}) = \sum_{l=1}^{L} \omega_{k-1}^{l} \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}^{l}, P_{k-1|k-1}^{l}) \tag{5}$$

where $Z_{k-1} \triangleq \{Z_{k-1}^1, \ldots, Z_{k-1}^N\}$ and $Z_{k-1}^i \triangleq \{z_1^i, \ldots, z_{k-1}^i\}$ denote the cumulative sets of measurements received from all sensor nodes and the ith sensor node, respectively. ω_{k-1}^l, $\hat{x}_{k-1|k-1}^l$ and $P_{k-1|k-1}^l$ are the weight, mean and covariance of the lth Gaussian component, respectively.

The time updated predictive distribution can now be obtained by

$$\begin{aligned} p(x_k|Z_{k-1}) &= \int p(x_k|x_{k-1}) p(x_{k-1}|Z_{k-1})\, dx_{k-1} \\ &= \sum_{l=1}^{L}\sum_{p=1}^{G_1} \omega_{k-1}^{l} \alpha_p \int \mathcal{N}(x_k, f(x_{k-1}), Q_{k-1,p}) \\ &\quad \times \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}^{l}, P_{k-1|k-1}^{l})\, dx_{k-1} \\ &= \sum_{l=1}^{L}\sum_{p=1}^{G_1} \omega_{k-1}^{l} \alpha_p \mathcal{N}(x_k; \hat{x}_{k|k-1}^{l,p}, P_{k|k-1}^{l,p}) \end{aligned} \tag{6}$$

where $\hat{x}_{k|k-1}^l$ and $P_{k|k-1}^l$ can be calculated by generating sigma points as in UKF [7].

After receiving all the sensor measurements $\{z_k^1, \ldots, z_k^N\}$ at time step k, the filtering distribution can be derived by

$$\begin{aligned} p(x_k|Z_k) &= \frac{1}{C} p(z_k^1, \ldots, z_k^N | x_k) p(x_k|Z_{k-1}) \\ &= \frac{1}{C} \sum_{l=1}^{L}\sum_{p=1}^{G_1}\sum_{q=1}^{G_2} \omega_{k-1}^{l} \alpha_p \beta_q \Lambda_q(z_k^1, \ldots, z_k^N) \mathcal{N}(x_k; \hat{x}_{k|k}^{l,p,q}, P_{k|k}^{l,p,q}) \end{aligned} \tag{7}$$

where C is a normalizing constant. The filtering estimates $\hat{x}_{k|k}^{l,p,q}$ and $P_{k|k}^{l,p,q}$ are the updated mean and the covariance matrix using the centralized UKF. $\Lambda_q(z_k^1, \ldots, z_k^N)$ is the centralized likelihood function defined by

$$\Lambda_q(z_k^1, \ldots, z_k^N) = \mathcal{N}(\overline{z}_k; \hat{\overline{z}}_{k|k-1}, \overline{R}_{k,q} + \overline{P}_{zz,k}) \tag{8}$$

with $\overline{z}_k = [(z_k^1)^T, \ldots, (z_k^N)^T]^T$ and $\overline{R}_{k,q} = \mathrm{diag}\{R_{k,q}^1, \ldots, R_{k,q}^N\}$. $\hat{\overline{z}}_{k|k-1}$ and $\overline{P}_{zz,k}$ are respectively calculated based on the fused measurements $\overline{z}_k$.

To implement the filtering distribution in a distributed manner, it can be seen from (7) that the filtering estimates as well as the weight update should be calculated in a distributed fashion. Note that the filtering estimates $\hat{x}_{k|k}^{l,p,q}$ and $P_{k|k}^{l,p,q}$ can be obtained by directly using the average-consensus strategy as the distributed UKF [8], and the remaining work is to distribute the weight update. To this end, we have

$$\Lambda_q(z_k^1, \ldots, z_k^N) \approx \prod_{i=1}^{N} \Lambda_q(z_k^i) \tag{9}$$

where $\Lambda_q(z_k^i) = \mathcal{N}(z_k^i; \hat{z}_{k|k-1}^i, R_{k,q}^i + P_{zz,k}^i)$, and $\hat{z}_{k|k-1}^i$ and $P_{zz,k}^i$ are respectively calculated based on the ith sensor measurement z_k^i. A simple interpretation for the effectiveness of this approximation has been shown in [24].

Then the weight ω_k^t at time step k can be represented by

$$\omega_k^t \propto \omega_{k-1}^l \alpha_p \beta_q \prod_{i=1}^{N} \Lambda_q(z_k^i), \quad t = 1, \ldots, LG_1G_2 \tag{10}$$

However, the average-consensus strategy cannot be applied for the weight update due to the existence of the product term in (10). To overcome this difficulty, we consider the natural logarithm transformation, i.e., we can define

$$\Delta_k^t \triangleq \ln \prod_{i=1}^{N} \Lambda_q(z_k^i) = \sum_{i=1}^{N} \ln \Lambda_q(z_k^i) \tag{11}$$

By treating $\delta_k^t = \ln \Lambda_q(z_k^i)$ as the input of an average-consensus filter [25], the output $\hat{\delta}_k^t$ can asymptotically track the value $(1/N)\Delta_k^t$. Thus, the weight ω_k^t can be calculated by

$$\omega_k^t \propto \omega_{k-1}^l \alpha_p \beta_q \exp(N\hat{\delta}_k^t), \quad t = 1, \ldots, LG_1G_2 \tag{12}$$

In other words, each sensor node can communicate the message δ_k^t with its neighbors to carry out the weight update. It is worth noticing that the Gaussian mixture filter suffers from the computational problem associated with the exponentially increasing of Gaussian components as time progresses. There are many strategies dealing with this problem such as the pruning scheme [26], the expectation maximization (EM) algorithm [27] and the resampling method [17]. In this work, a simple pruning scheme is adopted to resolve this problem by discarding the components with insignificant weights so that the number of Gaussian components keeps constant at each time step.

For clarity, the distributed GM-UKF is summarized in Algorithm 1.

Algorithm 1. Distributed GM-UKF.

1. Initialization for sensor node i: $\omega_0^{i,l}$, $\hat{x}_{0|0}^{i,l}$, $P_{0|0}^{i,l}$
2. **while** new measurement is received **do**
3. Time update

Generate weighted sigma points

$$\chi_{k-1|k-1}^{i,l,0} = \hat{x}_{k-1|k-1}^{i,l}, \quad W_0 = \frac{\kappa}{n+\kappa}$$

$$\chi_{k-1|k-1}^{i,l,s} = \hat{x}_{k-1|k-1}^{i,l} + \left(\sqrt{(n+\kappa)P_{k-1|k-1}^{i,l}}\right)_s, \quad W_s = \frac{1}{2(n+\kappa)}, \quad s = 1, \ldots, n$$

$$\chi_{k-1|k-1}^{i,l,s+n} = \hat{x}_{k-1|k-1}^{i,l} - \left(\sqrt{(n+\kappa)P_{k-1|k-1}^{i,l}}\right)_s, \quad W_s = \frac{1}{2(n+\kappa)}, \quad s = 1, \ldots, n$$

where κ is a scaling factor and W_s is the weight associated with the sth sigma point. The vector $(\sqrt{(n+\kappa)P_{k-1|k-1}^i})_s$ denotes either the sth row or the sth column of the matrix square root of $(n+\kappa)P_{k-1|k-1}^i$. In general if the matrix square root A of P is of the form $P = A^T A$, then the sigma points are formed from the rows of A. Otherwise, the columns of A are used if $P = AA^T$ [7].

Calculate the predicted mean and the covariance matrix

$$\chi_{k|k-1}^{i,l,s} = f(\chi_{k-1|k-1}^{i,l,s})$$

$$\hat{x}_{k|k-1}^{i,l,p} = \sum_{s=0}^{2n} W_s \chi_{k|k-1}^{i,l,s}$$

$$P_{k|k-1}^{i,l,p} = \sum_{s=0}^{2n} W_s [\chi_{k|k-1}^{i,l,s} - \hat{x}_{k|k-1}^{i,l,p}][\chi_{k|k-1}^{i,l,s} - \hat{x}_{k|k-1}^{i,l,p}]^T + Q_{k-1,p}$$

Calculatethe predicted measurement and the cross-correlation covariance matrix

$$\hat{z}^{i,l,p}_{k|k-1} = \sum_{s=0}^{2n} W_s h^i(\chi^{i,l,s}_{k|k-1})$$

$$P^{i,l,p}_{xz,k} = \sum_{s=0}^{2n} W_s[\chi^{i,l,s}_{k|k-1} - \hat{x}^{i,l,p}_{k|k-1}][h^i(\chi^{i,l,s}_{k|k-1}) - \hat{z}^{i,l,p}_{k|k-1}]^T$$

$$P^{i,l,p}_{zz,k} = \sum_{s=0}^{2n} W_s[h^i(\chi^{i,l,s}_{k|k-1}) - \hat{z}^{i,l,p}_{k|k-1}][h^i(\chi^{i,l,s}_{k|k-1}) - \hat{z}^{i,l,p}_{k|k-1}]^T$$

4. Consensus update
Definethe inputs of average-consensus filters

$$\Lambda_q(z^i_k) = \mathcal{N}(z^i_k; \hat{z}^{i,l,p}_{k|k-1}, R^i_{k,q} + P^{i,l,p}_{zz,k})$$

$$\delta^{i,l,p,q}_k = \ln \Lambda_q(z^i_k)$$

$$y^{i,l,p,q}_{k|k} = [P^{i,l,p}_{k|k-1}]^{-1} P^{i,l,p}_{xz,k} [R^i_{k,q}]^{-1} (z^i_k - \hat{z}^{i,l,p}_{k|k-1})$$

$$Y^{i,l,p,q}_{k|k} = [P^{i,l,p}_{k|k-1}]^{-1} P^{i,l,p}_{xz,k} [R^i_{k,q}]^{-1} [P^{i,l,p}_{xz,k}]^T [P^{i,l,p}_{k|k-1}]^{-1}$$

Consensus iteration: $\tau \to \tau + T_c$

$$\delta^{i,l,p,q}_k(\tau+1) = \delta^{i,l,p,q}_k(\tau) + \sum_{j=1}^{N} \gamma_{ij}(\tau)(\delta^{j,l,p,q}_k(\tau) - \delta^{i,l,p,q}_k(\tau))$$

$$y^{i,l,p,q}_{k|k}(\tau+1) = y^{i,l,p,q}_{k|k}(\tau) + \sum_{j=1}^{N} \gamma_{ij}(\tau)(y^{j,l,p,q}_{k|k}(\tau) - y^{i,l,p,q}_{k|k}(\tau))$$

$$Y^{i,l,p,q}_{k|k}(\tau+1) = Y^{i,l,p,q}_{k|k}(\tau) + \sum_{j=1}^{N} \gamma_{ij}(\tau)(Y^{j,l,p,q}_{k|k}(\tau) - Y^{i,l,p,q}_{k|k}(\tau))$$

whereT_c is the number of local iteration steps to achieve average-consensus.
5. Measurement update

$$\hat{x}^{i,l,p,q}_{k|k} = \hat{x}^{i,l,p}_{k|k-1} + N P^{i,l,p,q}_{k|k} \hat{y}^{i,l,p,q}_{k|k}$$

$$[P^{i,l,p,q}_{k|k}]^{-1} = [P^{i,l,p}_{k|k-1}]^{-1} + N \hat{Y}^{i,l,p,q}_{k|k}$$

where $\hat{y}^{i,l,p,q}_{k|k} = y^{i,l,p,q}_{k|k}(T_c)$ and $\hat{Y}^{i,l,p,q}_{k|k} = Y^{i,l,p,q}_{k|k}(T_c)$ are outputs of the above average-consensus filters, respectively.
6. Weight update and discard Gaussian terms with weak weights

$$\omega^{i,l,p,q}_k \propto \omega^{i,l}_{k-1} \alpha_p \beta_q \exp(N \hat{\delta}^{i,l,p,q}_k), \quad \sum_{l=1}^{L} \sum_{p=1}^{G_1} \sum_{q=1}^{G_2} \omega^{i,l,p,q}_k = 1$$

where $\hat{\delta}^{i,l,p,q}_k = \delta^{i,l,p,q}_k(T_c)$ is the output of the above average-consensus filter.
Sort the LG_1G_2 components according to their weights $\omega^{i,l,p,q}_k$ in descending order. Retain only the first L components and denote the weights and filtering estimates as $\omega^{i,l'}_k$ and $\hat{x}^{i,l'}_{k|k}$, $P^{i,l'}_{k|k}$.
7. Combination of filtering estimates

$$\hat{x}^i_{k|k} = \sum_{l'=1}^{L} \omega^{i,l'}_k \hat{x}^{i,l'}_{k|k}$$

$$P^i_{k|k} = \sum_{l'=1}^{L} \omega^{i,l'}_k [P^{i,l'}_{k|k} + (\hat{x}^{i,l'}_{k|k} - \hat{x}^i_{k|k})(\hat{x}^{i,l'}_{k|k} - \hat{x}^i_{k|k})^T].$$

8. **end while**

4. Simulation results

This section provides a numerical example to compare the performance of the proposed distributed filter with that of the centralized fusion filter and the distributed particle filter. We consider a two-dimensional radar tracking scenario in the presence of glint, in which the glint measurement noise is modeled as non-Gaussian noise.

Tracking model: Assume that the target moves in the $x-y$ plane according to the following dynamics model:

$$x_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} w_{k-1} \tag{13}$$

where T is the sampling period and $x_k = [\xi_k \; \dot{\xi}_k \; \eta_k \; \dot{\eta}_k]^T$ is the target state vector. $(\xi_k \; \eta_k)$ and $(\dot{\xi}_k \; \dot{\eta}_k)$ denote the position and the velocity components at time step k, respectively. The process noise w_{k-1} is assumed to be zero-mean white Gaussian with covariance matrix Q_{k-1}.

Twelve radar sensors are used to provide measurements of range and bearing, i.e.,

$$z^i_k = \begin{bmatrix} \sqrt{(\xi_k - x^i_0)^2 + (\eta_k - y^i_0)^2} \\ \arctan[(\eta_k - y^i_0)/(\xi_k - x^i_0)] \end{bmatrix} + \nu^i_k, \quad i = 1, 2, \ldots, 12 \tag{14}$$

where (x^i_0, y^i_0) is the position of the ith sensor, and ν^i_k is the measurement noise in the presence of glint. In our simulations, two models are used to describe the glint noise ν^i_k.

Case I: In [28], the glint noise ν^i_k is modeled by a mixture of two zero-mean Gaussian terms, where the outliers were represented by a zero-mean Gaussian with a larger covariance matrix, i.e.,

$$p(\nu^i_k) = (1-\epsilon)\mathcal{N}(\nu^i_k; 0, R^i_1) + \epsilon \mathcal{N}(\nu^i_k; 0, R^i_2) \tag{15}$$

with ϵ being the glint probability.

Case II: In [29], the glint noise ν^i_k is modeled by a mixture of a zero-mean Gaussian noise with high occurrence probability and a Laplacian noise with low occurrence probability, i.e.,

$$p(\nu^i_k) = (1-\epsilon)\mathcal{N}(\nu^i_k; 0, R^i_1) + \epsilon \mathcal{L}(\nu^i_k; 0, R^i_2) \tag{16}$$

where ϵ is the glint probability, and $\mathcal{L}$ denotes the Laplacian probability density function.

Simulation results: The optimal centralized fusion filter, which involves measurements from all the sensors, serves as the baseline algorithm and the proposed distributed filter is compared with it. To emphasize the improved accuracy of data fusion, the tracking performance using a single sensor is also presented. For performance evaluation, the root mean-square error (RMSE) in position is utilized, and the simulation results are obtained from 100 Monte Carlo runs.

The prior information is taken as follows for all the algorithms under consideration. The prior estimate is $\hat{x}_{0|0} = x_0 + x_{\text{bias}}$, where x_0 is the true state of the target and x_{bias} is a bias on the position drawn from a Gaussian distribution with mean $[\beta \; 0 \; \beta \; 0]^T$. The Gaussian mixture is composed of four equally weighted Gaussian distributions with the same covariance matrix $P_{0|0} = \text{diag}\{2^2; 0.1^2; 2^2; 0.1^2\}$.

In the simulations, the initial target state is given by $x_0 = [-40 \; 3 \; 10 \; 1]^T$. The tracking performance is tested for 100 time steps with sampling period $T = 1$ s. As shown in

Fig. 1, 12 sensors are located at (0 30), (0 60), (0 90), (80 30), (80 60), (80 90), (160 30), (160 60), (160 90), (240 30), (240 60) and (240 90). The communication topology between sensors is presented in Fig. 2. It can be seen that they are not fully connected. For the GM-UKF, the conditional distribution of the target state vector is assumed to be Gaussian mixture model of order $L=4$. In the UKF, the scaling parameter κ is set to -1 [7].

Case I: The glint probability is taken to be $\epsilon=0.1$ and the covariance matrix of w_{k-1} is $Q_{k-1}=\mathrm{diag}\{0.01;0.01\}$. The covariance matrices R_1^i and R_2^i are taken to be $\mathrm{diag}\{0.2^2;0.015^2\}$ and $\mathrm{diag}\{2^2;0.15^2\}$ for all sensors, respectively. The parameter β for determining the initial bias is taken to be $\beta=1$. For simplicity of notation, the filter using the ith sensor measurement and the centralized fusion filter without GM technique are shortly denoted by Si-UKF and CF-UKF, respectively. The centralized fusion filter and the distributed GM-UKF using the ith sensor measurement with GM technique are denoted by CF-GM-UKF and Si-DF-GM-UKF, respectively. The performance comparison with respect to RMSE in position is shown in Fig. 3. Firstly, it is obvious that the RMSE with data fusion is smaller than those with a single measurement alone and the CF-GM-UKF performs better than the CF-UKF. This is expected due to the fact that the measurement noise is modeled as Gaussian mixture. Secondly, to implement the distributed GM-UKF, the Metropolis weight and $T_c=45$ consensus iteration steps have been used to derive the averages. It can be observed that the average-consensus has been reached for each local filter. Specially, the RMSE in position for the Si-DF-GM-UKF and the CF-GM-UKF are almost identical. These results suggest that the tracking performance of the proposed distributed GM-UKF is comparable to the centralized fusion filter. It is worth mentioning that the study of accelerating consensus algorithms has become another challenging topic and has recently received great attention [30–33].

Case II: Following the same line of Case I, the simulation results are shown in Fig. 4. Similar conclusions can be drawn that the proposed filter is able to cope with non-Gaussian noise in a distributed manner based on the GM technique.

To illustrate the effect of initialization on the performance of the proposed filter, the parameter β is set to 1, 2 and 3, respectively. The corresponding RMSE in position is shown in Fig. 5. It can be observed that the distributed GM-UKF (DF-GM-UKF) is not sensitive to the initial bias.

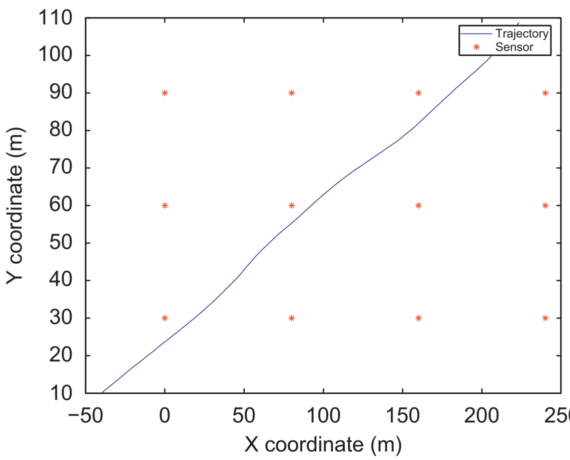

Fig. 1. Target tracking scenario with 12 sensors.

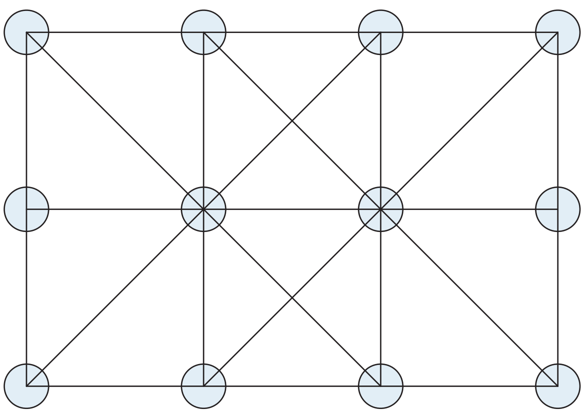

Fig. 2. Communication topology between sensors.

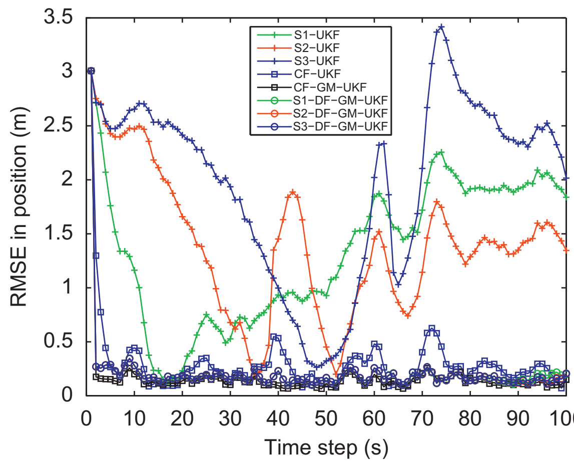

Fig. 3. Case I: performance comparison with UKF-based algorithms.

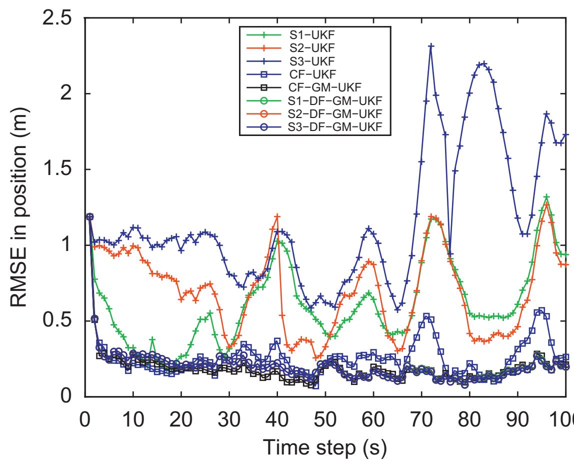

Fig. 4. Case II: performance comparison with UKF-based algorithms.

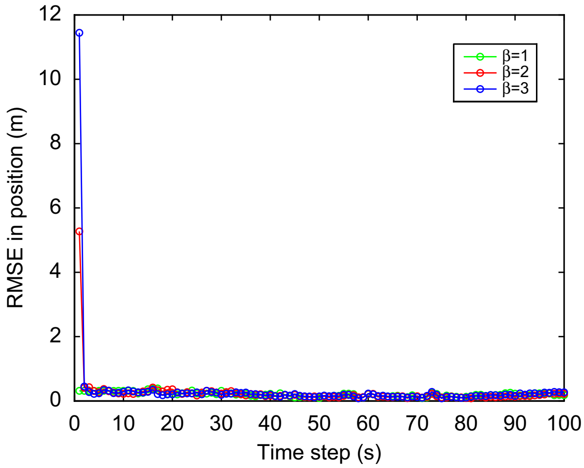

Fig. 5. Case II: RMSE versus initial bias.

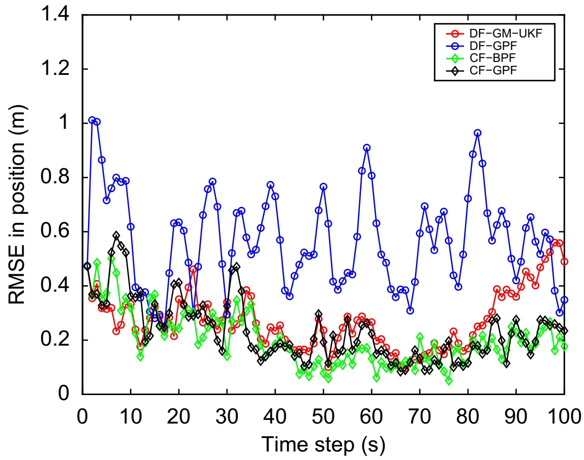

Fig. 7. Case II: performance comparison with PF-based algorithms.

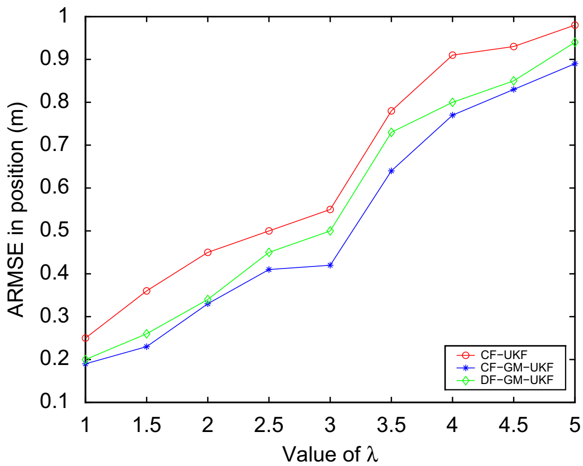

Fig. 6. Case II: ARMSE versus measurement noise.

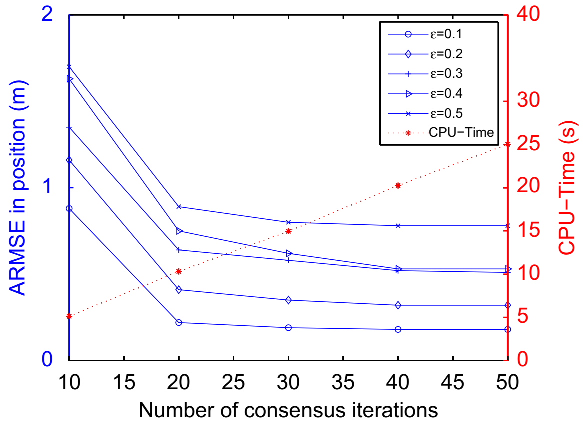

Fig. 8. Case II: ARMSE and CPU-Time versus number of consensus iterations.

However, the proposed filter might diverge if the bias is too large (e.g., $\beta > 5$). The average RMSE (ARMSE) in position with different levels of measurement noise is displayed in Fig. 6, where the variance of the Laplacian noise is enlarged by a scalar λ (i.e., $R^i_{2,\lambda} = \mathrm{diag}\{(2\lambda)^2;\ (0.15\lambda)^2\}$). The simulation results suggest that the performance is degraded when the level of the measurement noise is increased and the performance of the DF-GM-UKF is still close to that of the CF-GM-UKF. Moreover, they are better than the CF-UKF.

As shown in Fig. 7, the performance of the DF-GM-UKF is compared with particle filters including the centralized bootstrap particle filter (CF-BPF) with deterministic resampling scheme [9], the centralized Gaussian particle filter (CF-GPF) [34] and the distributed Gaussian particle filter (DF-GPF) with likelihood consensus [13]. The particle filters are implemented with 1000 particles and the number of consensus iterations T_c is set to 45 for distributed filters. The comparisons show that the CF-BPF and the CF-GPF have better performance than the DF-GM-UKF. However, the RMSE of the DF-GPF is higher than the DF-GM-UKF. From the simulations, it appears that the relative computational requirements (with respect to the DF-GM-UKF) for the CF-BPF, CF-GPF and DF-GPF are 4.2, 4.5 and 11.3, respectively. It is noted that the UKF-based algorithm has small computation time compared with particle filters and the computation time of the DF-GPF is the highest since the likelihood consensus iterations should be carried out for each particle.

Furthermore, we compute the ARMSE in position with respect to different numbers of consensus iteration and glint probabilities by averaging RMSE over all time instants and sensor nodes. The simulation results are shown in Fig. 8. From Fig. 8, it can be seen that the performance of the distributed GM-UKF is worse as the glint probability increases. This is expected since the covariance matrix of the Laplacian noise is larger than that of the Gaussian noise and higher occurrence probability of Laplacian noise may degrade the performance. On the other hand, the ARMSE of the distributed GM-UKF is lower as the number of consensus iteration increases but there is almost no improvement when T_c changes from 40 to 50. This is due to the fact that the consensus has been reached and the performance cannot be improved by more consensus iterations. The CPU-Time presented in Fig. 8 indicates that the processing time for each sensor node is linear with respect to the number of consensus iterations. As the latency and communication requirement are also related to the number of consensus iteration, the trade-off between tracking accuracy and communication requirement can be used in practical applications, i.e., the latency and the communication requirements should be tolerated if the tracking accuracy is an important issue, and vice versa.

5. Conclusion

In this paper, we propose a novel distributed GM-UKF for discrete-time nonlinear systems with non-Gaussian noise when the non-Gaussian noise can be approximated by a sum of weighted Gaussian components. The communication topology of the sensor network under consideration is assumed to be not fully connected and each sensor node can only communicate with its neighbors. The proposed distributed GM-UKF involves the computations of the local sufficient statistics and the mixture weights, which are resolved by using the dynamic average-consensus strategy. As each sensor node only requires the information exchange with its neighbors, the proposed filter is robust and scalable. Simulation results show that the performance of the proposed distributed filter is comparable to that of the centralized fusion filter which involves all the sensor measurements.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was jointly supported by the National 973 Program (2012CB821200) and the NSFC (61134005, 60921001, 90916024, 91116016).

References

[1] R. Carli, A. Chiuso, L. Schenato, S. Zampieri, Distributed Kalman filtering based on consensus strategies, IEEE Journal on Selected Areas in Communications 26 (4) (2008) 622–633.
[2] U.A. Khan, J.M.F. Moura, Distributing the Kalman filter for large-scale systems, IEEE Transactions on Signal Processing 56 (10) (2008) 4919–4935.
[3] W. Yu, G. Chen, Z. Wang, W. Yang, Distributed consensus filtering in sensor networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (6) (2009) 1568–1577.
[4] R. Olfati-Saber, J. Shamma, Consensus filters for sensor networks and distributed sensor fusion, in: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, Seville, Spain, December 2005, pp. 6698–6703.
[5] R. Olfati-Saber, Distributed Kalman filter with embedded consensus filters, in: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, Seville, Spain, December 2005, pp. 8179–8184.
[6] R. Olfati-Saber, Distributed Kalman filtering for sensor networks, in: Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA, December 2007, pp. 5492–5498.
[7] S.J. Julier, J.K. Uhlmann, H.F. Durrant-Whyte, A new method for nonlinear transformation of means and covariances in filters and estimators, IEEE Transactions on Automatic Control 45 (3) (2000) 477–482.
[8] W. Li, Y. Jia, Consensus-based distributed multiple model UKF for jump Markov nonlinear systems, IEEE Transactions on Automatic Control 57 (1) (2012) 230–236.
[9] B. Ristic, S. Aruplampalam, N. Gordon, Beyond the Kalman Filter: Particle Filter for Tracking Applications, Artech House, Boston, 2004.
[10] D. Gu, Distributed particle filter for target tracking, in: Proceedings of IEEE International Conference on Robotics and Automation, Roma, Italy, April 2007, pp. 3856–3861.
[11] D. Gu, J. Sun, Z. Hu, H. Li, Consensus based distributed particle filter in sensor networks, in: Proceedings of IEEE International Conference on Information and Automation, Changsha, China, June 2008, pp. 302–307.
[12] S. Farahmand, S.I. Roumeliotis, G.B. Giannakis, Particle filter adaptation for distributed sensors via set member-ship, in: Proceedings of IEEE International Conference Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, USA, March 2010, pp. 3374–3377.
[13] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, M. Rupp, Distributed Gaussian particle filtering using likelihood consensus, in: Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Prague, Czech Republic, May 2011, pp. 3756–3759.
[14] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, M. Rupp, Likelihood consensus: principles and application to distributed particle filtering, in: Proceedings of the 44th Asilomar Conference on Signals Systems and Computers, Pacific Grove, CA, November 2010, pp. 349–353.
[15] B.N. Oreshkin, M.J. Coates, Asynchronous distributed particle filter via decentralized evaluation of Gaussian products, in: Proceedings of the 13th International Conference on Information Fusion, Edinburgh, Scotland, July 2010, pp. 1–8.
[16] D.L. Alspach, H.W. Sorenson, Nonlinear Bayesian estimation using Gaussian sum approximation, IEEE Transactions on Automatic Control 17 (4) (1972) 439–448.
[17] J.H. Kotecha, P.M. Djuric, Gaussian sum particle filtering, IEEE Transactions on Signal Processing 51 (10) (2003) 2602–2612.
[18] I. Bilik, J. Tabrikian, MMSE-based filtering in presence of non-Gaussian system and measurement noise, IEEE Transactions on Aerospace and Electronic Systems 46 (3) (2010) 1153–1170.
[19] B. Safarinejadian, M.B. Menhaj, M. Karrari, Distributed variational Bayesian algorithms for Gaussian mixtures in sensor networks, Signal Processing 90 (4) (2010) 1197–1208.
[20] A. Mukherhee, A. Sengupta, Likelihood function modeling of particle filter in presence of non-stationary non-Gaussian measurement noise, Signal Processing 90 (6) (2010) 1873–1885.
[21] P.A. Bliman, G. Ferrari-Trecate, Average consensus problems in networks of agents with delayed communications, Automatica 44 (8) (2008) 1985–1995.
[22] M. Zhu, S. Martinez, Discrete-time dynamic average consensus, Automatica 46 (2) (2010) 322–329.
[23] B. Shen, Z. Wang, Y.S. Hung, Distributed H_∞-consensus filtering in sensor networks with multiple missing measurements: the finite-horizon case, Automatica 46 (10) (2010) 1682–1688.
[24] T. Vercauteren, X. Wang, Decentralized sigma-point information filters for target tracking in collaborative sensor networks, IEEE Transactions on Signal Processing 53 (8) (2005) 2997–3009.
[25] D. Kingston, R. Beard, Discrete-time average-consensus under switching network topologies, in: Proceedings of the American Control Conference, Minneapolis, MN, USA, June 2006, pp. 3551–3556.
[26] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Boston, 1999.
[27] D. Gu, Distributed EM algorithm for Gaussian mixtures in sensor networks, IEEE Transactions on Neural Networks 19 (7) (2008) 1154–1166.
[28] G. Hewer, R. Martin, J. Zeh, Robust preprocessing for Kalman filtering of glint noise, IEEE Transactions on Aerospace and Electronic Systems 23 (1) (1987) 120–128.
[29] I. Bilik, J. Tabrikian, Maneuvering target tracking in the presence of glint using the nonlinear Gaussian mixture Kalman filter, IEEE Transactions on Aerospace and Electronic Systems 46 (1) (2010) 246–262.
[30] E. Kokiopoulou, P. Frossard, Polynomial filtering for fast convergence in distributed consensus, IEEE Transactions on Signal Processing 57 (1) (2009) 342–354.
[31] T. Aysal, B. Oreshkin, M. Coates, Accelerated distributed average consensus via localized node state prediction, IEEE Transactions on Signal Processing 57 (4) (2009) 1563–1576.
[32] S. Sardellitti, M. Giona, S. Barbarossa, Fast distributed average consensus algorithms based on advection–diffusion processes, IEEE Transactions on Signal Processing 58 (2) (2010) 826–842.
[33] F. Xi, J. He, Z. Liu, Adaptive fast consensus algorithm for distributed sensor fusion, Signal Processing 90 (5) (2010) 1693–1699.
[34] J.H. Kotecha, P.M. Djuric, Gaussian particle filtering, IEEE Transactions on Signal Processing 51 (10) (2003) 2590–2601.