IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2019, accepted October 15, 2019, date of publication October 17, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948139

An Efficient Gaussian Sum Filter Based on Prune-Cluster-Merge Scheme

YANG XU[1], YANGWANG FANG[1,2], WEISHI PENG[1,3], AND YOULI WU[1]

[1]Department of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China
[2]Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China
[3]Department of Equipment Management and Support, People Armed Police Engineering University, Xi'an 710086, China

Corresponding author: Weishi Peng (peng_weishi@163.com)

This work was supported in part by the Major Program of the National Natural Science Foundation of China under Grant 61627901, in part by the National Natural Science Foundation of China under Grant 71801222 and Grant 61973253, in part by the National Science Foundation of Shaanxi Province of China under Grant 2018JQ6019, and in part by the National Postdoctoral Program for Innovative Talents under Grant BX201700104.

ABSTRACT The main problem for the state estimation with Gaussian mixture model is the exponentially growing number of Gaussian components. To solve this problem, an efficient Gaussian sum filter (GSF) based on the prune-cluster-merge (PCM) scheme-based Gaussian mixture reduction method is proposed. First, an adaptive weight-censored pruning strategy named as the j-th order statistical technique is presented to delete components with little contributes to the posterior distribution. Then, the Gauss clustering method is proposed to partition the remainder components into clusters based on the newly defined distribution similarity criterion. Integrating the acquired clusters with the covariance intersection algorithm, the components in the same cluster are merged into a standard Gaussian component by keeping the shape of the original distribution. Meanwhile, an extended integral square error cost function is constructed to optimize the performance of the cluster-merge operation. Finally, an efficient Gaussian sum filter is derived by combining the PCM scheme with extended Kalman filters. Numerical results show that the proposed filter can not only keep a better approximation to the original distribution with fewer Gaussian components comparing with the number-limited GSF and Runnalls's GSF, but achieve a higher cost-effectiveness than the particle filter.

INDEX TERMS Gaussian sum filter, Gaussian mixture, distribution similarity criteria, covariance intersection.

NOMENCLATURE

k	time instant
σ	standard deviation of the component
σ^*	confidence range
α	regularization constant
ρ	censoring threshold
$\Omega_{\bar{N}}$	parameter set of $\bar{N}$ components containing weights, means and covariances
N	component number after pruning operations
N_u	bound of the system on component numbers
$N_,$	cluster number after clustering operations
N_i^*	total component number of the i^*-th cluster
i_j^*	the j-th component in the i^*-th cluster
u_{k+1}^i	first moment of the i-th component
$\bar{u}_{k+1}^{i^*}$	first moment of the merged component for the i^*-th cluster
u_{ij}^{k+1}	Euclidean distance between the centers of the i^*-th and j-th component
$\bar{\omega}_{k+1}^{i^*}$	weight of the merged component for the i^*-th clust
$\omega_{k+1}^{i_m^*}$	weight of the m-th component in the i^*-th cluster
$d_{k+1}^{\sigma_i^*}$	the σ^* confidence range of the i-th component
$d_{k+1}^{i^*m}$	Euclidean distance between the m-th component and i^*-th cluster
c_{k+1}^p	the p-th component
C_{k+1}^i	the i-th cluster
$C_{k+1}^{i^*}$	the i-th cluster without overlapped components
$\bar{P}_{k+1}^{i^*}$	second moment of the merged component for the i^*- th cluster

The associate editor coordinating the review of this manuscript and approving it for publication was Yanbo Chen.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

$f(c_{k+1}\|C_{k+1}^{i^*})\sigma^*$	original distribution of the i^*-th cluster partitioned by σ^* confidence range
$q(c_{k+1}\|C_{k+1}^{i^*})\sigma^*$	estimated distribution of the i^*-th cluster partitioned by σ^* confidence range

I. INTRODUCTION

A Gaussian mixture (GM) approximation for the nonlinear non-Gaussian (NN) state probability density function (PDF) has received a great amount of attention in the context of Bayesian estimation [1]–[3]. For a nonlinear dynamic system with additive Gaussian noise, the first two moments of the Gaussian components are propagated through a linearized model, and the weight of the new component is set according to the prior weight and innovation. In the case where measurements are available on the first two moments, and weights are accordingly updated by Bayes rules to obtain an approximation of the posterior PDF, forming the so-called Gaussian Sum Filter (GSF) [4]. The GSF approximates the PDF using a weighted sum of Gaussian densities, which can effectively approximate any PDF as closely as desired.

The literature on nonlinear filtering using Gaussian mixture approximation approach is rich in theoretical work on the GSF [5], the mixture of Kalman filters [6] and particle GSF [7], [8], etc. In many applications such as target detection and tracking [9], speech enhancement [10] and image processing [11], the Gaussian mixture serves as an important probabilistic representation of the system. However, a serious restraint is the exponential growth of component numbers over time. To evolve an efficient Gaussian mixture reduction (GMR) approach, a pioneering work was done [4] and most practical implementations used the simplified probability densities to approximate the original PDF. The main focus was to select available Gaussian components based on their weights [12], and a pruning technique was proposed by setting thresholds [4], [9], which can retain any number of Gaussian components. Meanwhile, the joining and clustering algorithms [13] employed the *ad hoc* distance to measure the distance between two mixture components for the limit of components. Furthermore, Crouse presented a survey of GMR algorithms [14], in which enhanced West algorithm [15], constraint optimized weight adaptation [16], Runnall's algorithm [17] and GMR via clustering [18] were introduced and compared.

More recent studies by Akbar and Konstantinos [19] and Ganjac *et al.* [20] pointed that some applications, such as model validation [21], image retrieval [22], were much sensitive to the geometric shape of the GM. Accordingly, Arjovsky proposed a Wasserstein-distance-based GMR method with consideration of the unique properties of Wasserstein distance [23]. While, many applications such as [24] and target tracking [25], are more concerned with minimal change in the distribution shape of the GM. Most of the researchers engaged to improve the filter algorithm for better approaching the posterior PDF of the state, such as Fritsch investigated a smoothing method using GMM for multi-target tracking problem [26], and Cheol introduced a hybrid Bayesian network with Kullback-Leibler (KL) divergence to optimize the tracking accuracy with a bound on computation time [27]. However, there is little discussion about the GMR method in the view of keeping the original distribution shape. Inspired by this point, the objective of this paper is therefore to design a GMR algorithm for reducing the change in the distribution shape of the original GMs. To fulfill this task, a suboptimal GMR approach is proposed by integrating pruning, clustering and merging operations with nonlinear optimizations. The main contributions of this paper are summarized as follows:

(1) To remove the components with little contributions to the distribution of original GMs, a weight-censored pruning method termed as the j-th order statistical technique is applied according to a censoring threshold. By taking 3σ rule into account, the significant components can be remained efficiently.

(2) Different from most existing clustering methods, a Gauss clustering approach is proposed based on a newly defined distribution similarity criterion. To derive an optimal approximation performance, an extended integral square error (EISE) cost function is constructed for optimizing the confidence range, which is the key influencing factor to the clustering results.

(3) By applying the covariance intersection (CI) algorithm to the approximation problem, the components in the same cluster are merged into a standard Gaussian component for keeping it tractable for further processing steps.

The rest sections of this paper are organized as follows. In section II the conventional GSF is introduced. Section III derives a GMR approach based on the PCM scheme. In section IV, the confidence range is analysed, and simulations are conducted to validate the effectiveness of our proposed PCM scheme. Section V gives the conclusions and provides an outlook for the future work.

II. CONVENTIONAL GAUSSIAN SUM FILTER

In this section, the Gaussian mixture model is first introduced, which is the basis of the Gaussian sum filters; the conventional Gaussian sum filter integrated a bunch of extended Kalman filters is derived after, followed by the main issue for the GSFs.

A. GAUSSIAN MIXTURE MODEL

It is well known that the Gaussian mixture model (GMM) can effectively capture any PDF as closely as desired, and a Gaussian mixture PDF is a weighted sum of Gaussian components as follows.

$$f(x|\Omega_{\bar{N}}) \approx \sum_{i=1}^{\bar{N}} \breve{w}_i N(x|x^{(i)}, P^{(i)}) \tag{1}$$

where $\Omega_{\bar{N}} = \left\{\breve{w}_i, N(x|x^{(i)}, P^{(i)})\right\}_{i=1}^{\bar{N}}$ is the full mixture of components with the constraints $\sum_{i=1}^{\bar{N}} \breve{w}_i = 1,\ \forall \breve{w}_i \geq 0$.

Each Gaussian component satisfies the following density

$$N(x|\Omega_{\bar{N}}) = \frac{\exp(-\frac{1}{2}(x-x^{(i)})^T(P^{(i)})^{-1}(x-x^{(i)}))}{(2\pi)^{n/2}|P^{(i)}|^{1/2}} \quad (2)$$

where n means the dimension of x, and here we first assume x is a scalar.

Clearly, the reduced PDF is decided by the parameters of component number $\bar{N}$, the weight $\breve{w}_i$, the mean $x^{(i)}$, and the covariance $P^{(i)}$.

B. GAUSSIAN SUM FILTER

For the dynamic state space model, assume the prior probability densities of $p(x_k)$, $p(w_k)$ and $p(v_k)$ are expressed as follows, respectively.

$$\begin{cases} p(x_k) = \sum_{i=1}^{I_k} \alpha_k^{(i)} N(x_k|x_k^{(i)}, P_k^{(i)}) \\ p(w_k) = \sum_{l=1}^{L_k} \gamma_k^{(l)} N(w_k|w_k^{(l)}, Q_k^{(l)}) \\ p(v_k) = \sum_{j=1}^{J_k} \beta_k^{(j)} N(v_k|v_k^{(j)}, R_k^{(j)}) \end{cases}$$

$$s.t. \begin{cases} \sum_{i=1}^{I_k} \alpha_k^{(i)} = \alpha_k^{(i)} \geq 0 \\ \sum_{l=1}^{L_k} \gamma_k^{(l)} = \gamma_k^{(l)} \geq 0 \\ \sum_{j=1}^{J_k} \beta_k^{(j)} = \beta_k^{(j)} \geq 0 \end{cases} \quad (3)$$

where the subscript k represents the time instant, and I_k, L_k and J_k are the component number for the state, state noise and observation noise, respectively.

As all components satisfy Gaussian distributions, only the first two moments are demanded, and the extended Kalman filter (EKF) is employed as the basic nonlinear filter. In this manner, the estimation procedures can be described as follows.

(1) Time update process

$$x_{k+1|k}^{(i')} = f(x_k^{(i)}) + w_k^{(l)} \quad (4)$$

$$P_{k+1|k}^{(i')} = Q_k^{(l)} + F_{k+1}^{(i)} P_k^{(i)} (F_{k+1}^{(i)})^T \quad (5)$$

$$F_{k+1}^{(i)} = \left.\frac{\partial f}{\partial x}\right|_{x=x_k^{(i)}} \quad (6)$$

$$\phi_k^{(i')} = \alpha_k^{(i)} \gamma_k^{(l)}, \quad i' = 1, 2, ..., \xi_k \quad (7)$$

$$\xi_k = I_k L_k \quad (8)$$

where ξ_k is the component number after time update process.

(2) Measurement update process

$$x_{k+1}^{(i'')} = x_{k+1|k}^{(i')} + K_{k+1}^{(i'')}(z_{k+1} - h(x_{k+1|k}^{(i')}) - v_{k+1}^{(j)}) \quad (9)$$

$$P_{k+1}^{(i'')} = P_{k+1|k}^{(i')} - K_{k+1}^{(i'')} H_{k+1}^{(i')} P_{k+1|k}^{(i')} \quad (10)$$

$$H_{k+1}^{(i')} = \left.\frac{\partial h}{\partial x}\right|_{x=x_{k+1|k}^{(i')}} \quad (11)$$

$$K_{k+1}^{(i'')} = P_{k+1|k}^{(i')} (H_{k+1}^{(i')}(x_{k+1|k}^{(i')}))^T \cdot (R_{k+1}^{(j)} + H_{k+1}^{(i')} P_{k+1|k}^{(i')} (H_{k+1}^{(i')})^T)^{-1} \quad (12)$$

$$q(z_{k+1}^{(i'')}) = N(z_{k+1}; h(x_{k+1|k}^{(i')}) + v_{k+1}^{(j)}, R_{k+1}^{(j)} + H_{k+1}^{(i')} P_{k+1|k}^{(i')} (H_{k+1}^{(i')})^T) \quad (13)$$

$$w_{k+1}^{(i'')} = \psi_k^{(i'')} q(z_{k+1}^{(i'')}) \Big/ \sum_{i''=1}^{I_{k+1}} \psi_k^{(i'')} q(z_{k+1}^{(i'')}) \quad (14)$$

$$\psi_{k+1}^{(i'')} = \phi_k^{(i')} \beta_k^{(j)}, \quad i'' = 1, 2, ..., I_{k+1} \quad (15)$$

$$I_{k+1} = \xi_k J_k \quad (16)$$

where I_{k+1} is the component number after measurement update process.

A point estimate for state is taken to obtain the posterior mean and covariance, which are computed as

$$\hat{x}_{k+1} = \sum_{i''=1}^{I_{k+1}} w_{k+1}^{(i'')} x_{k+1}^{(i'')} \quad (17)$$

$$\hat{P}_{k+1} = \sum_{i''=1}^{I_{k+1}} \alpha_{k+1}^{(i'')} [P_{k+1}^{(i'')} + Q_{k+1}(Q_{k+1})^T] \quad (18)$$

where $Q_{k+1} = \hat{x}_{k+1} - x_{k+1}^{(i'')}$.

The GSF approximation for the conditional state PDF approaches the true conditional PDF under the assumption that there is a sufficient number of Gaussian components, and the covariances of all Gaussian components are small enough so that the linearization around the means are representative for the dynamic in the vicinity of the respective means [29]. However, the assumption that the state noise and/or measurement noise are expressed by the GMM makes the number of components constantly increase during propagation. Thus, an efficient GMR strategy is imperative.

III. PCM SCHEME-BASED GMR

In this section, an efficient GMR scheme is explored, which is divided into three operations: pruning, clustering and merging. Pruning strategy is introduced first, which is relatively independent of the other operations, but it affects the approximate performance. Then a novel criterion for clustering is proposed, followed by the CI merging algorithm. While, the criterion for clustering will be fine-tuned depending on the merging performance.

A. ADAPTIVE WEIGHT-CENSORED PRUNING

Clearly, the component number will increase exponentially when $L_k > 1$ or $J_k > 1$. While, the results indicate that most of the weights of components are too small to have significant impacts on the approximation results, which agrees with the conclusions drawn from [30]. Accordingly, a novel pruning approach called the j-th order statistical technique is proposed. Different from the existing weight-based pruning

approaches [4], [13], [31], this method removes the components in a reverse view, which is more propitious to retain the original distribution shape [32].

First, sort the weights of components in an ascending order

$$w_{k+1}^{(1)} \leq \cdots w_{k+1}^{(j)} \cdots \leq w_{k+1}^{(I_{k+1})} \tag{19}$$

where $w_{k+1}^{(j)}$ denotes the j-th smallest weight (the so-called j-th order statistic).

Then, define γ_n as

$$\gamma_n = \sum_{l=n}^{I_{k+1}} w_{k+1}^{(l)} \Big/ \sum_{l=1}^{I_{k+1}} w_{k+1}^{(l)} \tag{20}$$

Set $n_{\max}$ the maximum value of n satisfying the inequality

$$n_{\max} = \max_{n}(arg(\gamma_n \geq \rho)) \tag{21}$$

where ρ denotes a censoring threshold.

Maintain the largest $I_k - n_{\max} + 1$ components, and then compare with the upper boundary of the number N_u. Hence the remainder component number N can be computed as

$$N = \min\left(I_{k+1} - n_{\max} + 1, N_u\right) \tag{22}$$

Notice that the number N is associated with ρ, which is determined according to accuracy requirements and computation capabilities. Unreasonable selections may cause a degeneration of Gaussian components, such as, if $\rho = 0$, only the component with the maximum weight will be remained. While, a larger ρ makes the reduction operation be out of effect. Hence, after a thorough investigation according to the prior information, the 3σ rule is introduced to assist with determination of ρ. For a Gaussian distribution, the values within two standard deviations explicate about 95.4%; and within three standard deviations account for about 99.7%. Taking the computation efficiency into account, the 2σ region is used, therefore, we take $\rho = 0.9544$.

B. GAUSS CLUSTERING

In general, the remainder component number N is large if ρ approaches 1. To further reduce the component number, a Gauss clustering method is defined to identify and cluster the components, which are similar in distribution properties.

To measure the distance between the components, a center distance matrix is first defined as

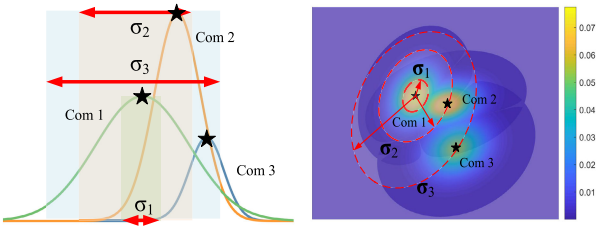
$$\Pi_{k+1} = \begin{bmatrix} u_{k+1}^{11} & u_{k+1}^{12} & \cdots & u_{k+1}^{1N} \\ u_{k+1}^{21} & u_{k+1}^{22} & \cdots & u_{k+1}^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k+1}^{N1} & u_{k+1}^{N2} & \cdots & u_{k+1}^{NN} \end{bmatrix} \tag{23}$$

where u_{k+1}^{ij} represents the Euclidean distance between the center of the i-th and j-th component, and $u_{k+1}^{ij} = |u_{k+1}^{i} - u_{k+1}^{j}|,\ i, j = 1, 2 \ldots N$; u_{k+1}^{i} represents the center of the i-th component. Since the component satisfies Gaussian distribution, the center can be treated as the first moment.

To identify the similar components with the center distance matrix, we motivated by [33] put forward a principal definition of the distribution similarity criterion as follows.

Definition 1: If the centers of two components are within each other's confidence range, we call these two components similarity in the distribution sense.

For easier interpretation, two examples are given in Fig.1.



(a) One-dimensional case (b) Two-dimensional case

FIGURE 1. A schematic depiction of the distribution similarity criterion.

In Fig.1, the black start denotes the first moment, and the red line is the confidence scope. Parameter σ^* represents the confidence range. From the graph above we can see Com 2 and Com 3 play different roles to Com 1 with the change of σ^*. If $\sigma^* = \sigma_1$, we have $u_{12}, u_{13} > \sigma^*$; while, if $\sigma^* = \sigma_2$, then $u_{12} < \sigma^*$, $u_{13} > \sigma^*$; if $\sigma^* = \sigma_3$, we get $u_{12}, u_{13} < \sigma^*$. Observe that as σ^* increases, the similar component with Com 1 expand, and this conclusion also holds for the two-dimensional case.

In order to quickly identify the similar components, a judgment matrix is then defined as

$$\Upsilon_{k+1} = \begin{bmatrix} r_{k+1}^{11} & r_{k+1}^{12} & \cdots & r_{k+1}^{1N} \\ r_{k+1}^{21} & r_{k+1}^{22} & \cdots & r_{k+1}^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k+1}^{N1} & \cdots & \cdots & r_{k+1}^{NN} \end{bmatrix} \tag{24}$$

where $r_{k+1}^{ij} = sign(\min\{d_{k+1}^{\sigma_i^*} - |u_{k+1}^{ij}|, d_{k+1}^{\sigma_j^*} - |u_{k+1}^{ij}|\})$ and $d_{k+1}^{\sigma_i^*}$ is the scope of the i-th component's σ^* confidence range.

Note that Υ_{k+1} has the following properties:

(1) Υ_{k+1} is a symmetric matrix, namely $r_{k+1}^{ij} = r_{k+1}^{ji}$;

(2) The diagonal element of Υ_{k+1} is 1, namely $r_{k+1}^{ii} = 1$;

(3) The matrix Υ_{k+1} is a -1−1 matrix, namely $r_{k+1}^{ij} \in \{-1, 1\}$. If $r_{k+1}^{ij} = 1$, then it means the i-th component intersects with the j-th component. Otherwise, it means one component deviates from the other's confidence range at least.

Based on these properties, looping through Υ_{k+1} by rows, then the similar components can be identified by

$$C_{k+1}^{i} = \left\{ c_{k+1}^{i}, c_{k+1}^{j} \middle| r_{k+1}^{ij} = 1,\ j = 1, ..., N \right\} \tag{25}$$

where C_{k+1}^{i} is the i-th cluster, which is a set of similar components, and c_{k+1}^{i} is the i-th component.

At this time, the preliminary cluster number is identical to the component number for the existence of the overlapped components, which is assigned to more than one clusters simultaneously. To avoid the reutilization problem caused by the overlapped components and renew the clusters, a restriction that any component can only pertain to one cluster is put forward, and the fine-tuning method for the clusters is given as follows.

Step 1: Combine the same cluster into one among N clusters.

Step 2: Classify the remaining clusters into following situations.

$$C_{k+1}^{i} \cap C_{k+1}^{j} = \begin{cases} \emptyset & \textit{Situation } 1 \\ \{c_{k+1}^{m} \ldots\} & \textit{Situation } 2 \\ C_{k+1}^{j} & \textit{Situation } 3 \end{cases} \tag{26}$$

For different situations, the corresponding solution is presented as follows.

Situation 1: Keep two clusters invariant.

Situation 2: Proceed with the procedures as follows.

Step 2.1: Eliminate the overlapped component c_{k+1}^{m} from the clusters.

$$\begin{cases} C_{k+1}^{i^*} = \{c_{k+1}^{p} | c_{k+1}^{p} \in C_{k+1}^{i}, c_{k+1}^{p} \notin \{c_{k+1}^{m}\}\} \\ C_{k+1}^{j^*} = \{c_{k+1}^{q} | c_{k+1}^{q} \in C_{k+1}^{j}, c_{k+1}^{q} \notin \{c_{k+1}^{m}\}\} \end{cases} \tag{27}$$

Step 2.2: Measure the distance between c_{k}^{m} and the above renew clusters based on the nearest neighbor logic.

The distance between a component and a cluster is defined as

$$d_{k+1}^{i^*m} = \min[|u_{k+1}^{i_1^*} - u_{k+1}^{m}|, \ldots, |u_{k+1}^{i_{n_i^*}^*} - u_{k+1}^{m}|] \tag{28}$$

where $u_{k+1}^{i_t^*}, t = 1, \ldots, n_i^*$ is the center of the t-th component in the i^*-th cluster; n_i^* is the number of components in the i^*-th cluster.

Step 2.3: Reassign the overlapped component and update the clusters.

If $d_{k+1}^{i^*m} \leq d_{k+1}^{j^*m}$, then c_{k+1}^{m} will be reassigned to $C_{k+1}^{i^*}$. Otherwise, it will be clustered to $C_{k+1}^{j^*}$.

Step 2.4: Repeat Step 2.1-Step 2.3 until no overlapped components exist.

Situation 3: Preserve C_{k+1}^{j} and its complement as independent clusters.

Step 3: Repeat the above procedures for all clusters in pairs.

At this moment, the number of clusters are assumed to change from N to N'. In order to achieve a desirable approximation performance, the number of clusters still needs control. Meanwhile, as σ^* plays an essential role on clustering results, and therefore, we put forward the following EISE cost function to derive the optimal σ^*.

$$\min_{\sigma^*} J_S = \frac{1}{2} \sum_{i=1}^{N'} \int [f(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*} - q(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*}]^2 dc_{k+1} + \alpha e^{N'/N} \tag{29}$$

where $f(c_{k+1}|C_{k+1}^{i^*})_{\sigma*}$ represents the original PDF of the i^*-th cluster with

$$f(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*} = \sum_{j=1}^{n_i^*} w_{k+1}^{i_j^*} N(x_{k+1}|u_{k+1}^{i_j^*}, P_{k+1}^{i_j^*}) \tag{30}$$

$q(c_{k+1}|C_{k+1}^{i^*})_{\sigma*}$ represents the approximate PDF of the i^*-th cluster, which is obtained according to the merging algorithm. α is the regularization constant, and N' represents the cluster number.

As seen from Eq.(29), it is suggested that the more clusters we form, the better the approximation accuracy achieves, but the larger regularization term is, so the regularization term can trade accuracy of the approximation for the number of clusters classified effectively. For ease of comprehension, the process for determining the number of clusters is typified in the following flowchart.

In Fig.2, the termination condition means the change of the adjacent EISE is less than a small value. Theoretically, the number of the clusters cannot be definitely confirmed before implementing the PCM operation, which is only determined by the distribution property of each Gaussian component. The number of clusters is invariant in different time instants or scenarios. For convenience, we assume the cluster number after PCM operations is N' in this paper.

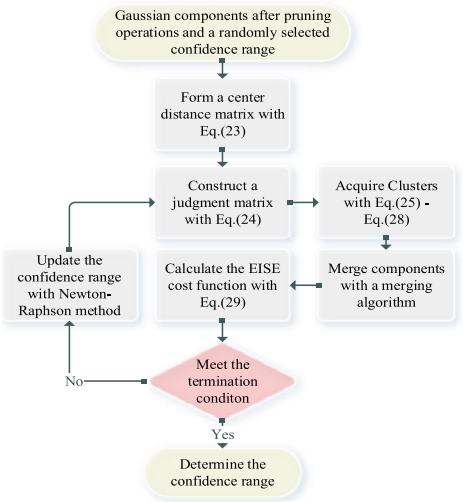


FIGURE 2. The flow chart of determining the number of clusters.

To minimize the EISE cost function, we first simplify the integral term as

$$\int (f(c_{k+1}|C_{k+1}^{i^*})_{\sigma*} - q(c_{k+1}|C_{k+1}^{i^*})_{\sigma*})^2 dc_{k+1} = J_{hh} - 2J_{hr} + J_{rr} \tag{31}$$

where

$$\begin{cases} J_{hh} = \int (f(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*})^2 dc_{k+1} \\ J_{hr} = \int (f(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*}) \cdot q(c_{k+1}|C_{k+1}^{i^*})_{\sigma*} dc_{k+1} \\ J_{rr} = \int q(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*}^2 dc_{k+1} \end{cases}$$

As the merged component satisfies Gaussian distribution, it can be described by

$$q(c_{k+1}|C_{k+1}^{i^*})_{\sigma^*} = \bar{\omega}_{k+1}^{i^*} \cdot N(x_{k+1}|\bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i^*}) \tag{32}$$

where $\left\{\bar{\omega}_{k+1}^{i^*}, \bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i^*}\right\}$ represent the weight, mean and variance of the merged component for the i^*-th cluster, respectively.

Substituting Eq.(32) into Eq.(31) yields

$$\begin{cases} J_{hh} = \sum_{m=1}^{n_i^*}\sum_{n=1}^{n_i^*} \omega_{k+1}^{i_m^*}\omega_{k+1}^{i_n^*} \int N(x_{k+1}|u_{k+1}^{i_m^*}, P_{k+1}^{i_m^*})\cdot \\ \qquad N(x_{k+1}|u_{k+1}^{i_n^*}, P_{k+1}^{i_n^*})dx_{k+1} \\ J_{hr} = \sum_{m=1}^{n_i^*} \omega_{k+1}^{i_m^*}\bar{\omega}_{k+1}^{i^*} \int N(x_{k+1}|u_{k+1}^{i_m^*}, P_{k+1}^{i_m^*})\cdot \\ \qquad N(x_{k+1}|\bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i^*})dx_{k+1} \\ J_{rr} = (\bar{\omega}_{k+1}^{i^*})^2 \int N(x_{k+1}|\bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i})\cdot \\ \qquad N(x_{k+1}|\bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i^*})dx_{k+1} \end{cases} \tag{33}$$

Remark 1: To derive the indispensable equations in this paper, one Gaussian identity that is the product of two Gaussian PDFs can be simplified as

$$N(x|\mu_1, P_1) \cdot N(x|\mu_2, P_2) = \beta N(x|\mu_3, P_3) \tag{34}$$

where

$$\begin{cases} \beta = N(\mu_1|\mu_2, P_1 + P_2) \\ P_3 = (P_1^{-1} + P_2^{-1})^{-1} \\ \mu_3 = P_3\left(P_1^{-1}\mu_1 + P_2^{-1}\mu_2\right) \end{cases}$$

and β is the scaling factor.

From the above equation we realize if the integral operation is over the entire space of a Gaussian distribution, then only the scaling factors will leave. For this point, Eq.(33) can be simplified as

$$\begin{cases} J_{hh} = \sum_{m=1}^{n_i^*}\sum_{n=1}^{n_i^*} w_{k+1}^{i_m^*} w_{k+1}^{i_n^*} \cdot \\ \qquad N(u_{k+1}^{i_m^*}|u_{k+1}^{i_n^*}, P_{k+1}^{i_m^*} + P_{k+1}^{i_n^*}) \\ J_{hr} = \sum_{m=1}^{n_i^*} w_{k+1}^{i_m^*}\bar{w}_{k+1}^{i^*} \cdot \\ \qquad N(u_{k+1}^{i_m^*}|\bar{u}_{k+1}^{i^*}, P_{k+1}^{i_m^*} + \bar{P}_{k+1}^{i^*}) \\ J_{rr} = (\bar{w}_{k+1}^{i^*})^2 N(\bar{u}_{k+1}^{i^*}|\bar{u}_{k+1}^{i^*}, 2\bar{P}_{k+1}^{i^*}) \end{cases} \tag{35}$$

To this end, Eq.(31) consists of the sum of similarity measures of all pairs of two components from the same cluster, and similarity measures of all pairs of two components from the merged component, and the sum of similarity measures of all pairs of one component from the cluster and one component from the merged component. Implement this result to all clusters, and add a regularization term, then the value of EISE will be obtained. In order to optimize the above clustering operation, we then give a principal assumption on the selection of σ^*.

Assumption 1: Parameter σ^* is assumed to be associated with the second moment of the component, namely $\sigma^* = \lambda\sigma$.

According to the above assumption, σ^* can be screened out with respect to the inherent distribution feature of components. In the end, we adopt the classic Newton-Raphson method to derive the optimal λ, and the details about this iteration process can be found in [34].

C. COVARIANCE INTERSECTION MERGING

According to the above clustering method, the remainder components are assumed to be partitioned into N' clusters. As suggested in [35], it is possible to combine many components into a single one without seriously affecting the approximation accuracy. Pursuing this idea, we intend to merge components belonging to the same cluster into one component by keeping the distribution shape of original GMs.

First of all, the merging question is equivalent to the following, that is, there are two pieces of information, labelled A and B, to be merged together to yield an output C, which is a general type of data fusion problem. Meanwhile, the estimators obtained in the GMM satisfy the consistency and unbiasedness [36], and in this manner, CI algorithm is adopted to get a consistent estimate of two or more estimates in the same cluster. In general, CI algorithm takes a convex combination of mean and covariance estimates. When the local error variance $P^{(i)}$ is exactly known, but the cross-covariance $P^{(ij)}$ $(i \neq j)$ is unknown, using the CI merging algorithm, which is formulated as

$$P^{(i,j)} = w(P^{(i)})^{-1} + (1-w)(P^{(j)})^{-1} \tag{36}$$

$$(P^{(i,j)})^{-1}x^{(i,j)} = w(P^{(i)})^{-1}x^{(i)} + (1-w)(P^{(j)})^{-1}x^{(j)} \tag{37}$$

where $x^{(i,j)}$ and $P^{(i,j)}$ represent the first and second moments of the merged component, respectively.

The update equation is consistent for different selections of w proved in [37], and different w results in different covariance ellipses of the merging component, which is shown in Fig.3.

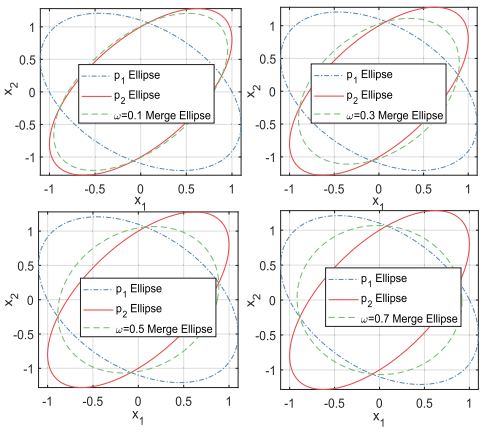


FIGURE 3. Covariance ellipse with different *w*.

As illustrated in Fig.3, the parameter w affects the weights assigned to $x^{(i)}$ and $x^{(j)}$. Different w can be used to optimize the update equation for different performance criteria, for instance, minimizing the trace of $P^{(i,j)}$. A convex cost function with respect to w has only one distinct optimum in the range $0 \leq w \leq 1$. Virtually any optimization strategy can be used, ranging from Newton-Raphson to sophisticated semi-definite and convex programming techniques, which can minimize almost any norm.

Likewise, the conclusions above can be extended to the i^*-th cluster [36] as follows.

$$\left(\bar{P}_{k+1}^{i^*}\right)^{-1} = \sum_{j=1}^{n_i^*} w_{k+1}^{i_j^*} (P_{k+1}^{i_j^*})^{-1} \tag{38}$$

$$\bar{u}_{k+1}^{i^*} = \bar{P}_{k+1}^{i^*} (\sum_{j=1}^{n_i^*} w_{k+1}^{i_j^*} (P_{k+1}^{i_j^*})^{-1} u_{k+1}^{i_j^*}) \tag{39}$$

where the optimal weighted coefficients $w_{k+1}^{i_j^*}$ $(j = 1, \ldots, n_i^*)$ can be determined by minimizing the performance index with the constraints $\sum_{j=1}^{n_i^*} w_{k+1}^{i_j^*} = 1, w_{k+1}^{i_j^*} \in [0, 1]$.

$$\min_{w} tr(\bar{P}_{k+1}^{i*}) = \min tr\{[\sum_{j=1}^{n_i^*} w_{k+1}^{i_j^*} P_{k+1}^{i_j^*}]^{-1}\} \tag{40}$$

This is a nonlinear optimization problem with constraints in Euclidean space, which can be solved by "fmincon" function in MATLAB toolbox. Here we utilize the normalized weights instead of searching for the optimal weighted coefficients.

$$\bar{w}_{k+1}^{i_j^*} = w_{k+1}^{i_j^*} \Big/ (1 - \sum_{m=1}^{I_{k+1}-N+1} w_{k+1}^{(m)}) \tag{41}$$

The merging procedure holds for all the clusters, and in the end, N' merged components are propagated as the initial inputs, $\Omega_{N'} = \{\bar{w}_{k+1}^{i^*}, N(x_{k+1}|\bar{u}_{k+1}^{i^*}, \bar{P}_{k+1}^{i^*})\}_{i^*=1}^{N'}$, where $\bar{w}_{k+1}^{i^*}$ is the sum of component weights in the i^*-th cluster.

To this end, combining the PCM scheme with EKFs, an efficient GMR-GSF is constructed, and the procedures are illustrated in Tab.1.

IV. NUMERICAL EXAMPLE AND SIMULATION

To begin with we will provide the underlying purposes for the following experiments. In this section, we divide the experiments into four parts. Firstly, the proper searching scope of σ^* is investigated with three typical situations. Then, in order to verify the approximation performance of PCM scheme-based GMR algorithm (PCM-GMR) and better visualize the performance with Runnalls's GMR algorithm (R-GMR) [17], which measures the similarity based on the KL divergence, and number-limited GMR algorithm (NL-GMR) [12], two static scalar cases are considered. Following this, a state estimation problem in one dimension and a range-bearing tracking problem in four dimensions are presented to further test the effectiveness and efficiency of our proposed algorithm.

TABLE 1. Procedures of PCM-GSF.

Input: I_{k+1} Gaussian components, censoring threshold ρ, upper boundary of the component number N_u, regularization constant α;
1. Initialization: Obtain I_{k+1} components with the means, variances and weights, which are derived from Eq.(9), Eq.(10) and Eq.(14), respectively.
2. Weight-Censored Pruning: According to the censoring threshold ρ, select significant components with Eq.(19)-Eq.(21), and determine the numbers according to Eq.(22).
3. Reset Weights: Reset the weights of the remainder components with Eq.(41).
4. Form Clusters: Identify similar components by a given σ^* within a pre-determined scope with Eq.(23)-Eq.(25).
5. Renew Clusters: Reassign the overlapped component according to Eq.(27)-Eq.(28), and update the clusters.
6. Merge Components: Merge components in the same cluster into a Gaussian component by using CI algorithm with Eq.(38)-Eq.(39).
7. Optimize the Confidence Range: Apply the classic Newton-Raphson method to update the confidence range σ^* by minimizing Eq.(29) until it meets the termination condition.
8. Propagate Components: Propagate the obtained N' Gaussian components as the initial inputs for the next time recursive process with Eq.(3)- Eq.(18).

Besides, the complexities of the algorithms are compared by using the relative computation time of our MATLAB implementation central processing unite time (CPU-time) on Intel Core I5-4590 3.3GHz processor and 8GB RAM. Meanwhile, the regularization constant α is set to be 0.001, and ρ is 0.9544 for the PCM operations.

A. SEARCHING SCOPE ANALYSIS

1) UNIMODAL PDF CASE

First we give four components with following parameters.

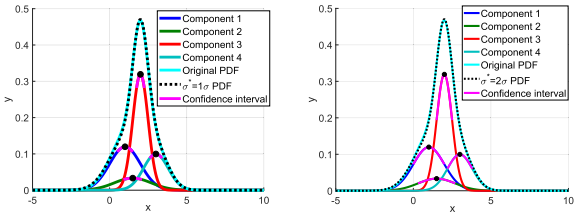
$$c_1 = \{0.3, N(x|1, 1^2)\}, \quad c_2 = \{0.1, N(x|1.5, 1.2^2)\}$$
$$c_3 = \{0.4, N(x|2, 0.5^2)\}, \quad c_4 = \{0.2, N(x|3, 0.8^2)\}$$

The approximation results are shown in Fig.4.

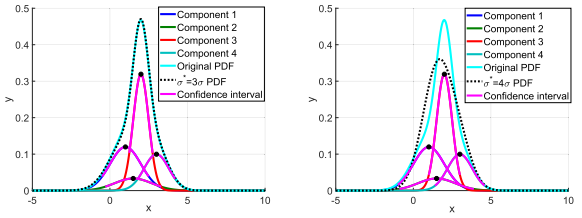
In Fig.4, Component 1 and 2 are considered similar and merged when $\sigma^* > 2\sigma$, and Component 3 is clustered into them when $\sigma^* = 4\sigma$. We note that the approximation accuracy is positive when $\sigma^* < 3\sigma$. While, as σ^* gets larger, more components are identified as similar ones, which makes the degeneration of the distribution diversity and leads to a negative result. The corresponding results are displayed in Tab.2.

TABLE 2. Results for approximating unimodal PDF.

Confidence range	Final number	EISE	CPU-time(s)
σ^*=1σ	4	0.0272	0.014
σ^*=2σ	3	0.0221	0.016
σ^*=3σ	3	0.0221	0.016
σ^*=4σ	2	0.0348	0.017



(a) σ^*=1σ approximate situation (b) σ^*=2σ approximate situation



(c) σ^*=3σ approximate situation (d) σ^*=4σ approximate situation

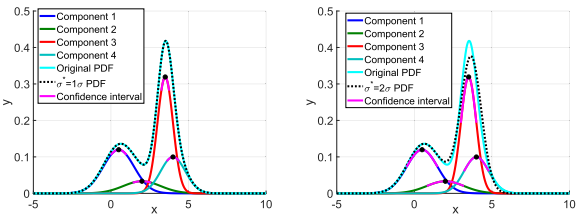
FIGURE 4. **Approximated unimodal PDF by four Gaussian components.**

Tab.2 shows that the cluster number decreases with σ^* increasing. While, the less the number of clusters is, the longer execution time it takes for more components need to be merged in the cluster. However, in comparison with the time spent for propagation, it is worthwhile to cost a little time for GMR operations. Besides, it should be noted that the EISE rises first and then falls. Considering the monotonicity of the EISE cost function, the optimal confidence range may locate within [1-3]σ region.

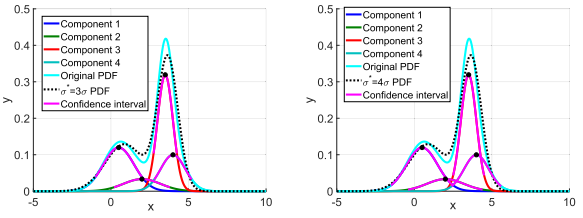
2) BIMODAL PDF CASE

The parameters of four components are given as follows, and the approximate curves are displayed in Fig.5.

$$c_1 = \{0.3, N(x|0.5, 1^2)\}, \quad c_2 = \{0.1, N(x|2, 1.2^2)\}$$
$$c_3 = \{0.4, N(x|3.5, 0.5^2)\}, \quad c_4 = \{0.2, N(x|4, 0.8^2)\}$$



(a) σ^*=1σ approximate situation (b) σ^*=2σ approximate situation



(c) σ^*=3σ approximate situation (d) σ^*=4σ approximate situation

FIGURE 5. **Approximated bimodal PDF by four Gaussian components.**

In Fig.5(a), no pair of components are similar enough to be clustered. When $\sigma^* = 2\sigma$, Component 3 and 4 are treated as similar in the distribution sense and then merged into a Gaussian component. Once the confidence range exceeds 3σ, Component 1 and 2 are classified into another cluster, which makes the approximation accuracy get worse, especially at the peak points. Tab.3 provides the approximation results.

It can be seen from the data in Tab.3 that the EISE starts to increase when σ^* exceeds 2σ, which illustrates that the optimal σ^* may be within [1-3]σ region.

TABLE 3. **Results for approximating bimodal PDF.**

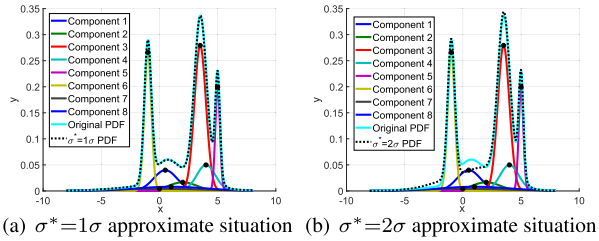
Confidence range	Final number	EISE	CPU-time(s)
σ^*=1σ	4	0.0272	0.014
σ^*=2σ	3	0.0229	0.016
σ^*=3σ	2	0.0285	0.018
σ^*=4σ	2	0.0285	0.018

3) MULTI-MODAL PDF CASE

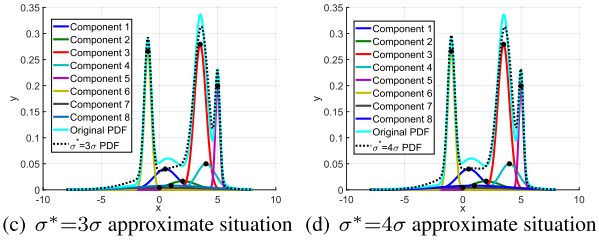
The parameters of eight components are presented as follows.

$$c_1 = \{0.1, N(x|0.5, 1)\}, \quad c_2 = \{0.05, N(x|2, 1.2^2)\}$$
$$c_3 = \{0.35, N(x|3.5, 0.5^2)\} \quad c_4 = \{0.1, N(x|4, 0.8^2)\}$$
$$c_5 = \{0.1, N(x|5, 0.2^2)\}, \quad c_6 = \{0.2, N(x|-1, 0.3^2)\}$$
$$c_7 = \{0.04, N(x|0, 4^2)\}, \quad c_8 = \{0.06, N(x|1, 3^2)\}$$

Fig.6 displays the approximate curves with different σ^*.



(a) σ^*=1σ approximate situation (b) σ^*=2σ approximate situation



(c) σ^*=3σ approximate situation (d) σ^*=4σ approximate situation

FIGURE 6. **Approximated multi-modal PDF by eight Gaussian mixtures.**

In Fig.6, Component 1 and 8 are identified as similar components and merged into a standard Gaussian component when $\sigma^* = 2\sigma$, and Component 3 and 4 are then clustered into another cluster and merged into a component when $\sigma^* = 3\sigma$. For this multi-modal case, the approximation accuracy to the peak gets worse with σ^* increasing. The approximation results are displayed in Tab.4.

As shown in Tab.4, the CPU-Time changes little in comparison with the other two cases, which indicates that the

TABLE 4. Results for approximating multi-modal PDF.

Confidence range	Final number	EISE	CPU-time(s)
σ^*=1σ	8	0.0240	0.015
σ^*=2σ	7	0.0238	0.017
σ^*=3σ	6	0.0314	0.019
σ^*=4σ	6	0.0314	0.018

PCM-GMR algorithm holds for complex situations for the high computation efficiency. Besides, from the EISE shown in Tab.4 we can see that the optimal confidence range may locate within [1,3]σ region.

As discussed above, the searching scope is determined from 1σ to 3σ. Through simulation analysis, the efficiency is improved by 2.4 times through setting the search scope in advance. For that reason, we initialize σ^* with 1.5σ in the subsequent parts.

B. STATIC SCENARIO ANALYSIS

In this part, we compare the performance of PCM-GMR with NL-GMR and R-GMR in two static scalar cases. The initial numbers of components are set to be 30 and 100, respectively. Meanwhile, the approximation performances are evaluated based on ISE, normalized ISE (NISE), EISE and CPU-time, and the first two indexes are described as follows.

(1) Integral Squared Error

The ISE is the most used index for GMR, which is described by

$$J_{ISE}(\Omega_{N'}) = \int_x (f(x|\Omega_I) - f(x|\Omega_{N'}))^2 dx \tag{42}$$

where I means the original number of components.

(2) Normalized Integral Squared Error

The NISE for GMR has been extensively researched by Petrucci [38], which can only change between zero and one,

$$J_{NISE}(\Omega_{N'}) = \frac{\int_x (f(x|\Omega_I) - f(x|\Omega_{N'}))^2 dx}{\int_x f(x|\Omega_I)^2 dx + \int_x f(x|\Omega_{N'})^2 dx} \tag{43}$$

1) BIMODAL SCENARIO BY 30 COMPONENTS

To better visualize the performance of GMR algorithms, the parameters are generated randomly with

$$\sum_{i=1}^{30} w_i = 1, \quad w_i \in [10^{-2}, 0.2],\ x_i \in [-2, 4],\ \sigma_i \in [0.4, 1]$$

and the upper boundary of component numbers for NL-GMR is set to be 20, and the goal of R-GWR is to utilize 20 components to approximate original distribution. Then the approximation results are displayed in Fig.7.

As shown in Fig.7, the pruning method, NL-GMR, discarding components only according to the weight is inferior to the other sophisticated GMR methods, especially at the peak position. At the same time, we note that PCM-GMR almost has the similar performance with R-GMR, but the former performs a little than the latter at the peak points.

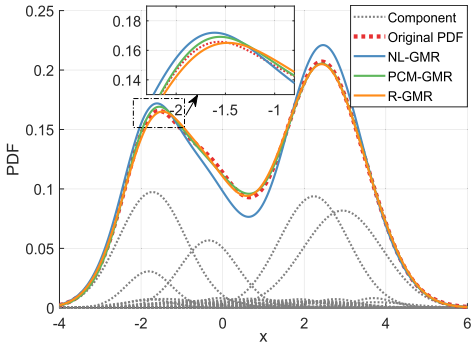


FIGURE 7. Approximation comparison when $I = 30$.

TABLE 5. Performance comparison when $I = 30$.

Index	NL-GMR	PCM-GMR	R-GMR
ISE	7.1e-2	2.5e-3	3.4e-3
NISE	2.5e-3	8.8e-5	1.2e-4
EISE	2.9e-3	1.7e-3	1.9e-3
CPU-time (s)	2e-3	1.1e-2	2.5e-2
Remaining Number	15	13	15

The corresponding results are provided in Tab.5, and from it we note that PCM-GMR performs the best for ISE, NISE and EISE indexes, followed by R-GMR, then NL-GMR. Moreover, PCM-GMR utilizes the least number of components when $\sigma^* = 1.15\sigma$. Although NL-GMR costs the minimum CPU-time, but it ignores to keep the distribution shape, which leads to a negative approximation accuracy in the end. While, R-GMR, which merges similar components in pairs according to the KL divergence, is more computationally demanding than the other competitive algorithms.

2) MULTI-MODAL SCENARIO BY 100 COMPONENTS

In this part, we consider a multi-modal scenario by using 100 components, and their parameters are limited by

$$\sum_{i=1}^{100} w_i = 1, \quad w_i \in [10^{-3}, 0.1],\ x_i \in [-7, 6],\ \sigma_i \in [0.3, 2]$$

and the number boundary for NL-GMR is 60, and the desirable component number for R-GMR is 60. With the above parameters, Fig.8 shows the performance of different algorithms.

It can be clearly seen from Fig.8 that the approximation accuracy of NL-GMR is the worst among the competitive algorithms, and PCM-GMR approaches to the original distribution with the best accuracy, especially at the peak positions. While, a slight decrease appears at the peak positions for R-GMR.

Tab.6 summarizes the performance indexes for comparison. From Tab.6 it is observed that PCM-GMR is superior to the competitive algorithms for the smaller ISE, NISE and EISE by using fewer components, which illustrates the high efficiency of the PCM scheme. Although PCM-GMR and R-GMR cost longer execution time than NL-GMR, the timeless for them is still in acceptable ranges.

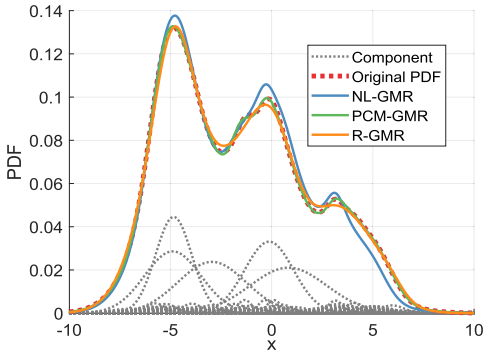


FIGURE 8. Approximation comparison when $I = 100$.

TABLE 6. Performance comparison when $I = 100$.

Index	NL-GMR	PCM-GMR	R-GMR
ISE	3.3e-2	4.2e-4	6.4e-3
NISE	1.9e-3	2.5e-5	3.9e-4
EISE	2.8e-3	1.8e-3	2.0e-3
CPU-time (s)	3e-3	3.1e-2	5.9e-2
Final Number	60	48	60

C. STATE ESTIMATION FOR A SCALAR SYSTEM

To test the nonlinearity effects on the performance of PCM-GSF with the competitive algorithms, in the following, we present a scalar measurement jump dynamic system where the measurement equation jumps to a linear equation within a special time interval. In addition, a more challenging competitive algorithm, particle filter (PF), is introduced with 300 particles, and R-GMR and NL-GMR apply EKFs as the basic nonlinear filters, and then we call them R-GSF and NL-GSF for short, respectively. All the performances are evaluated over 15 monte carlo (MC) runs, and the root mean square error (RMSE) is selected as the estimation accuracy index, which is described by [39], [40]

$$RMSE = \sqrt{\frac{1}{M_c}\sum_{j=1}^{M_c}\left(x_{k+1}^{j} - \hat{x}_{k+1}^{(j)}\right)^2} \tag{44}$$

where x_{k+1}^{j} and $\hat{x}_{k+1}^{(j)}$ represent the true and estimated values in the j-th MC running, respectively. M_c is the total number of MC running.

1) SIMULATION SCENARIO AND FILTER INITIALIZATION

Consider the measurement jump system with one-dimensional state is

$$x_{k+1} = \frac{x_k}{2} + \frac{25x_k}{1+x_k^2} + 8\cos(1.2k) + w_k \tag{45}$$

where the process noise w_k is described as Glint noise [41] and its PDF is constructed by Gaussian PDF and Laplace PDF, i.e.

$$f_t(x) = (1-\varepsilon) f_g(x) + \varepsilon f_l(x) \tag{46}$$

where $f_t(\cdot)$, $f_g(\cdot)$ and $f_l(\cdot)$ represents Glint noise, Gaussian noise and Laplace noise, respectively. ε means the glint frequency of the state. $f_l(\cdot)$ is formulated as

$$f_l(x) = \frac{1}{\eta}\exp\left(\frac{-|x|}{\eta}\right) \tag{47}$$

where η is the covariance of $f_l(x)$ and bigger than the covariance of $f_g(\cdot)$.

The measurement equation is given as follows

$$z_{k+1} = \begin{cases} \dfrac{x_{k+1}^2}{20} + v_{k+1} & 0 \le k < 80 \\ 3x_{k+1} - 4 + v_{k+1} & 80 \le k < 120 \end{cases} \tag{48}$$

where the measurement noise v_{k+1} is described by Gaussian PDF with $\mu = 2, \sigma = 3$. The initial state $p(x_0)$ is given by a sum of five Gaussian PDFs

$$\begin{aligned} p(x_0) = {} & 0.2N(x_0|-2, 10) + 0.2N(x_0|-1, 10) \\ & + \ldots + 0.2N(x_0|0, 10) + 0.2N(x_0|1, 10) \\ & + 0.2N(x_0|2, 10) \end{aligned} \tag{49}$$

Besides, a three-component Gaussian sum approximation of the Glint distribution by the spilt-merge incremental learning (SMILE) model [42] is denoted as

$$\begin{aligned} p(w_0) = {} & 0.29N(w_0|2.14, 0.72) + \ldots + 0.18N(w_0|7.45, 0.72) \\ & + 0.53N(w_0|4.31, 2.29) \end{aligned} \tag{50}$$

2) PERFORMANCE ANALYSIS

The RMSE curves for this jump system are plotted in Fig.9.

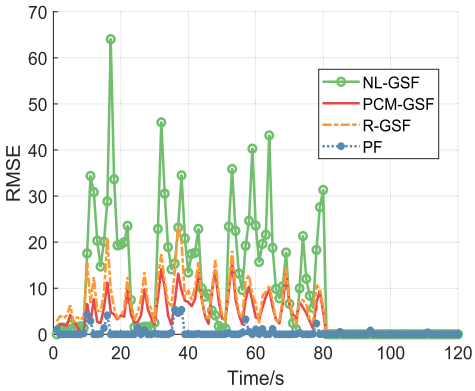


FIGURE 9. Comparison of RMSE in the measurement jump system.

As shown in Fig.9, as the limitation of EKFs, the RMSE of GSFs has weaker convergence results than PF before time instant 80. While, it is still promising to adopt more advanced nonlinear filters with GMMs. Seen from the variation amplitude of the RMSE curves for GSFs, it suggests that estimation accuracy for PCM-GSF is much better than NL-GSF, and is smoother than R-GSF in this dynamic NN system. In addition, all curves converge uniformly to a small value after time instant 80, which illustrates that the nonlinearity of the system has significant influences on the estimation performance.

The variations of the component numbers for the GSFs are compared in Fig.10, and for ease of visualization, we sample the data every five time instants.

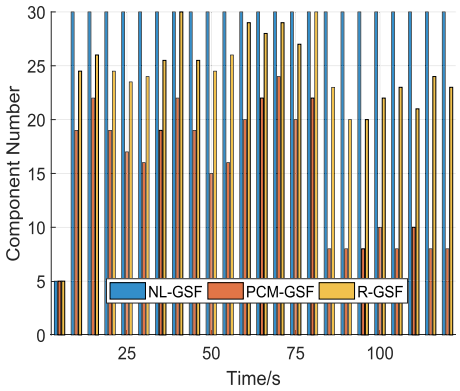


FIGURE 10. Comparison of number in the measurement jump system.

From Fig.10 we can see that when the nonlinearity of the system gets weak, the component numbers of PCM-GSF and R-GSF change over time, while NL-GSF still keeps it constant once reaching the upper boundary. Actually, more components are similar in the weak NN system, which makes the similarity-based GMR algorithm merge more components. We note that the component number of R-GSF is larger than that of PCM-GSF in the process, which illustrates that our proposed PCM-GMR algorithm is more efficient than KL divergence-based GMR algorithm.

Tab.7 shows the performance comparison in average RMSE (ARMSE), average EISE (AEISE) and CUP-time.

TABLE 7. Performances for the measurement jump system.

Method	ARMSE	AEISE	CPU-time(s)
NL-GSF	8.3394	0.287	8.25
PCM-GSF	2.0426	0.104	5.95
R-GSF	3.2812	0.184	6.84
PF	0.8763	-	23.14

Analyzing the data in Tab.7, we note that PCM-GSF outperforms NL-GSF and R-GSF in all performance indexes. Undeniably, PF achieves the best accuracy, but it takes much more time than others. While, PCM-GSF nearly fulfills the same task by using a quarter of execution time than PF at the cost of 1.2 ARMSE, and R-GSF is even more cost-effective than PF. Taking the efficiency into account, PF can only be applied in off-line systems, but PCM-GSF and R-GSF are more suitable for real-time systems.

The variation of the optimal confidence range over 15 MC runs is shown in Fig.11. From it we can see that the magnitude of the change of σ^* gets smaller when the nonlinearity turns weak, which illustrates that the nonlinearity of the system has significant impacts on σ^* .

D. RANGE-BEARING SCENARIO ANALYSIS

In order to test the effectiveness of PCM-GSF in a high dimensional system, a range-bearing scenario in the

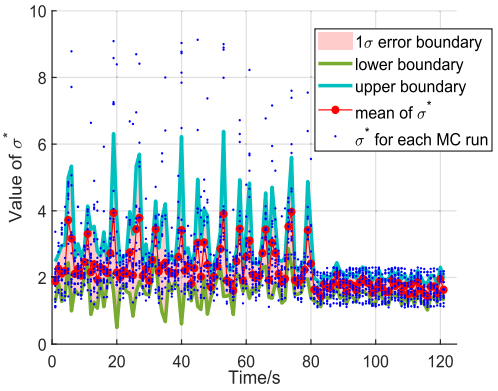


FIGURE 11. Variation of σ^* in the measurement jump system.

2-D plane [43] is presented. The target follows a constant velocity (CV) model, and the state vector contains the position and velocity in both X and Y directions, respectively. It is worth noting that our proposed PCM-GSF only considers the position information of the target, so the clustering and merging operations actually only deal with two-dimensional information here.

The nonlinear state space model is presented as

$$X_{k+1} = F \cdot X_k + G \cdot W_k \tag{51}$$

with

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix}$$

$$Z_k = h(X_k) + V_k \tag{52}$$

where

$$h(X_k) = \begin{bmatrix} \sqrt{(X_k^1)^2 + (X_k^3)^2} \\ \tan^{-1}(X_k^3 / X_k^1) \end{bmatrix}$$

and the state vector X_k in a Cartesian space is described by

$$X_k = \begin{bmatrix} X_k^1 \\ X_k^2 \\ X_k^3 \\ X_k^4 \end{bmatrix} = \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix} \tag{53}$$

Z_k corresponds to the set of noisy observations (range and bearing), and w_k is the state noise assumed to be Gaussian noise, zero-mean with variance-covariance matrix $Q = 0.175 \cdot diag([1, 1])$. V_k is the glint noise, and the glint frequency $\varepsilon = 0.02$. $dt = 0.01$ denotes the sampling period, and $T = 10$ denotes the simulation time. F and h are referred to the state transition matrix and observation transition matrix, respectively.

Besides, we consider the target only takes a 70 m/s^2 acceleration in Y direction. The initial state $X_0 = (1000\ m,\ 240\ m/s,\ 4000\ m,\ 0\ m/s)^T$, and the initial covariance

P_0=diag([120^2 m^2, 15^2 $(m/s)^2$, 100 m^2, 20^2 $(m/s)^2$]). X_0 and V_k are assumed to be represented by two and four Gaussian components, respectively, and their parameters are given by

$$c_1 = \{0.4, N(X|X_1, P_1)\}, \quad c_2 = \{0.6, N(X|X_2, P_2)\}$$
$$c_3 = \{0.3, N(V|V_1, R_1)\}, \quad c_4 = \{0.2, N(V|V_2, R_2)\}$$
$$c_5 = \{0.25, N(V|V_3, R_3)\}, \quad c_6 = \{0.25, N(V|V_4, R_4)\}$$

where

$$V_1 = (150\ m, 0.005\ rad)^T, \quad V_2 = (180\ m, 0.01\ rad)^T$$
$$V_3 = (85\ m, 0.009\ rad)^T, \quad V_4 = (100\ m, 0.015\ rad)^T$$
$$R_1 = diag([100^2\ m^2, 0.02^2\ rad^2])$$
$$R_2 = diag([80^2\ m^2, 0.02^2\ rad^2])$$
$$R_3 = diag([120^2\ m^2, 0.015^2\ rad^2])$$
$$R_4 = diag([110^2\ m^2, 0.01^2\ rad^2])$$
$$X_1 = (1000\ m, 220\ m/s, 4000\ m, 10\ m/s)^T$$
$$X_2 = (1000\ m, 190\ m/s, 4000\ m, 0)^T$$
$$P_1 = diag([150^2\ m^2, 10^2(m/s)^2, 100\ m^2, 30(m/s)^2])$$
$$P_2 = diag([120^2\ m^2, 8^2(m/s)^2, 90\ m^2, 10(m/s)^2])$$

The estimated posterior PDF of the target position in X direction from time instant 50 to 55 are shown in Fig.12.

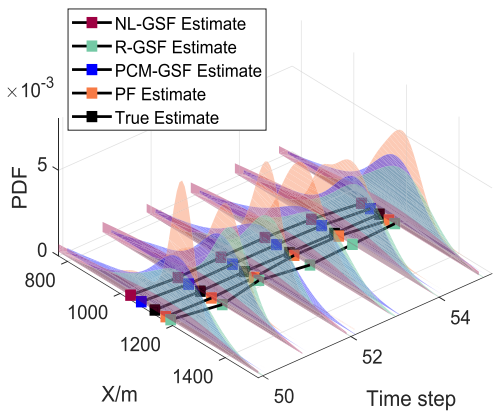


FIGURE 12. The estimated posterior PDF of the state.

Fig.12 shows that the positions estimated by PF approach to the true estimates much closer than other algorithms, and the posterior PDFs of the positions estimated by PCM-GSF are more and more coincident with these of PF, which illustrates that PCM-GSF has a strong self-regulation ability for approximating the true distribution. While, the posterior PDFs of R-GSF and NL-GSF change little over time, especially NL-GSF.

The estimation performances are displayed in Tab.8. Comparing the results obtained from the estimation process, the advantages of PF in estimation accuracy is obvious for the large amounts of particles. While, PCM-GSF also performs a positive result with a high efficiency. Meanwhile, we note that PF runs almost six times longer than PCM-GSF and four times longer than R-GSF. From the analysis above, it suggests that our proposed PCM scheme can efficiently approximate the posterior distribution in this dynamic four-dimensional case.

TABLE 8. Performances for the measurement jump system.

Method	ARMSE (m)	AEISE	CPU-time(s)
NL-GSF	112.3	0.327	0.06
PCM-GSF	58.2	0.043	0.28
R-GSF	93.6	0.064	0.45
PF	33.1	-	1.72

V. CONCLUSION

In this paper we propose a PCM-GSF to approximate the original posterior distribution with reduced components in the NN systems. For the scalar case, the PCM-GSF can achieve a better approximation accuracy of the PDF than NL-GSF and R-GSF, especially at the peak points. Moreover, the distribution similarity-based GMR can perform more efficiently than KL divergence-based GMR in reducing components. Comparing with PF, PCM-GSF can utilize almost a quarter of CPU-time by only sacrificing 1.2 ARMSE. When the nonlinearity changes weak, the performance of the GSFs can be significantly improved. For the dynamic high-dimensional case, PCM-GSF can approximate the true posterior distribution with a strong self-regulation ability, better than NL-GSF and R-GSF. Although the estimation accuracy need improve for the high-dimensional state, our proposed PCM-GSF is still more promising and suitable for applications with high real-time requirements comparing with NL-GSF, R-GSF and PF.

Future work includes extending the PCM scheme to a higher-dimensional system for a realistic implementation, and another effort will be spent on faster identifying similar components based on the distribution similarity criterion.

ACKNOWLEDGMENT

The authors are grateful to the reviewers for the precious suggestions and feedbacks for improving this paper and to the Editors for their meticulous processing.

REFERENCES

[1] A. G. Wills, J. Hendriks, C. Renton, and B. Ninness, "A Bayesian filtering algorithm for Gaussian mixture models," May 2017, *arXiv:1705.05495*. [Online]. Available: https://arxiv.org/abs/1705.05495

[2] Y. Hu and C. Han, "A Bayesian approach to track multiple extended targets using particle filter for nonlinear system," *Math. Problems Eng.*, vol. 2018, May 2018, Art. no. 7424538.

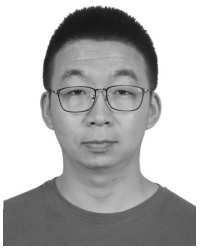
[3] Z. Liu, S. Chen, H. Wu, and K. Chen, "Robust student's t mixture probability hypothesis density filter for multi-target tracking with heavy-tailed noises," *IEEE Access*, vol. 6, pp. 39208–39219, Jul. 2018.

[4] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[5] L. Pishdad and F. Labeau, "A new reduction scheme for Gaussian sum filters," May 2014, *arXiv:1405.3164*. [Online]. Available: https://arxiv.org/abs/1405.3164

[6] S. S. Andreas and J. L. Rolf, "An iterative version of the adaptive Gaussian mixture filter," *Comput. Geosci.*, vol. 18, nos. 3–4, pp. 579–595, Aug. 2014.

[7] D. Raihan and S. Chakravorty, "Particle Gaussian mixture filters-I," *Automatica*, vol. 98, pp. 331–340, Dec. 2018.
[8] D. Raihan and S. Chakravorty, "Particle Gaussian mixture filters-II," *Automatica*, vol. 98, pp. 341–349, Dec. 2018.
[9] J. Zhang and Y. Liu, "Single maneuvering target tracking in clutter Based on multiple model algorithm with Gaussian mixture reduction," *J. Appl. Res. Technol.*, vol. 11, no. 5, pp. 641–652, Oct. 2013.
[10] B. B. Jiang, Z. Y. Li, and H. H. Chen, "Latent topic text representation learning on statistical manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5643–5654, Nov. 2018.
[11] J. I. Fernández-Michelli, M. Hurtado, J. A. Areta, and C. H. Muravchik, "Unsupervised classification algorithm based on EM method for polarimetric SAR Images," *ISPRS J. Photogram. Remote Sens.*, vol. 117, pp. 56–65, Jul. 2016.
[12] S. Ali-Loytty, "Efficient Gaussian mixture filter for hybrid positioning," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Monterey, CA, USA, May 2008, pp. 831–837.
[13] D. J. Salmond, "Mixture reduction algorithms for target tracking," in *Proc. IEE Colloq. State Estimation Aerosp. Tracking Appl.*, London, U.K., Dec. 1989, pp. 1–4.
[14] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at Gaussian mixture reduction algorithms," in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, IL, USA, Jul. 2011, pp. 1–8.
[15] K. C. Chang and W. Sun, "Scalable fusion with mixture distributions in sensor networks," in *Proc. 11th Int. Conf. Control, Autom., Robot., Vis.*, Singapore, Dec. 2010, pp. 1251–1256.
[16] H. D. Chen, K. C. Chang, and C. Smith, "Constraint optimized weight adaptation for Gaussian mixture reduction," *Proc. SPIE*, vol. 7697, Apr. 2010, Art. no. 76970N.
[17] A. R. Runnalls, "Kullback-leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.
[18] D. Schieferdecker and M. F. Huber, "Gaussian mixture reduction via clustering," in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, WA, USA, Jul. 2009, pp. 1536–1543.
[19] A. Assa and K. N. Plataniotis, "Wasserstein-distance-based Gaussian mixture reduction," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1465–1469, Oct. 2018.
[20] B. Gaujac, I. Feige, and D. Barber, "Gaussian mixture models with Wasserstein distance," Jun. 2018, *arXiv:1806.04465*. [Online]. Available: https://arxiv.org/abs/1806.04465
[21] A. Halder and R. Bhattacharya, "Probabilistic model validation for uncertain nonlinear systems," *Automatica*, vol. 50, no. 8, pp. 2038–2050, Aug. 2014.
[22] Y. Rubner, C. Tomasi, and L. J. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval*. Norwell, MA, USA: Kluwer, 2000.
[23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 214–223.
[24] Y. Li, J. Li, J. Qi, and L. Chen, "Robust cubature Kalman filter for dynamic state estimation of synchronous machines under unknown measurement noise statistics," *IEEE Access*, vol. 7, pp. 29139–29148, 2019.
[25] H. Zhang, X. Zhou, Z. Wang, H. Yan, and J. Sun, "Adaptive consensus-based distributed target tracking with dynamic cluster in sensor network," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1580–1591, May 2019.
[26] G. S. Fritsch and K. J. DeMars, "Smoothing for nonlinear multi-target filters with Gaussian mixture approximations," in *Proc. Space Flight Mech. Meeting*, Kissimmee, FL, USA, Jan. 2018, p. 0473.
[27] C. Y. Park, K. B. Laskey, P. C. G. Costa, and S. Matsumoto, "Gaussian mixture reduction for time-constrained approximate inference in hybrid Bayesian networks," *Appl. Sci.*, vol. 9, 2055, May 2019.
[28] H. Y. Zhu and Q. Z. Zhai, "A global optimal Gaussian mixture reduction approach based on integer linear programming," *Chin. J. Electron.*, vol. 22, no. 4, pp. 763–768, Oct. 2013.
[29] M. L. Psiaki, "Gaussian mixture nonlinear filtering with resampling for mixand narrowing," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 5499–5512, Nov. 2016.
[30] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. R. Statist. Soc. B*, vol. 62, no. 3, pp. 493–508, 2000.
[31] T. Li, J. Prieto, and J. M. Corchado, "A short revisit of nonlinear Gaussian filters: State-of-the-art and some concerns," in *Proc. Int. Conf. Ubiquitous Wireless Broadband*, Nanjing, China, Oct. 2016, pp. 1–4.
[32] X. Liu, K. Wang, P. Zhu, and K. Yang, "Lmproved JPDA algorithm with measurements adaptively censored," in *Proc. Int. Conf. Ind. Control Electron. Eng.*, Xi'an, China, Aug. 2012, pp. 207–211.
[33] X. Lv, Y. Ma, X. He, H. Huang, and J. Yang, "CciMST: A clustering algorithm based on minimum spanning tree and cluster centers," *Math. Problems Eng.*, vol. 2018, Dec. 2018, Art. no. 8451796.
[34] G. Ireneusz and G. Krzysztof, "Control of dynamics of the modified Newton–Raphson algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 67, pp. 76–99, Feb. 2019.
[35] C. Hennig, "Methods for merging Gaussian mixture components," *Adv. Data Anal. Classification*, vol. 4, no. 1, pp. 3–34, Apr. 2010.
[36] C. Z. Han, H. Y. Zhu, and Z. S. Duan, *Multi-Source Information Fusion*, 2nd ed. Beijing, China: Tsinghua Univ. Press, 2006.
[37] M. Liggins, D. L. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2001.
[38] D. J. Petrucci, "Gaussian mixture reduction for Bayesian target tracking in cluster," M.S. thesis, Dept. Eng. Manage., Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 2005.
[39] W.-S. Peng, Y.-W. Fang, and D. Chai, "Enhanced dynamic error spectrum for estimation performance evaluation in target tracking," *Optik*, vol. 127, no. 8, pp. 3943–3949, 2016.
[40] W. Peng, Y. Li, Y. Fang, Y. Wu, and Q. Li, "Radar chart for estimation performance evaluation," *IEEE Access*, vol. 7, no. 1, pp. 113880–113888, 2019.
[41] H.-W. Li and J. Wang, "Particle filter for manoeuvring target tracking via passive radar measurements with glint noise," *IET Radar, Sonar Navigat.*, vol. 6, no. 3, pp. 180–189, Mar. 2012.
[42] B. Konstantinos and E. L. Isaac, "Split–merge incremental learning (SMILE) of mixture models," in *Proc. Int. Conf. Artif. Neural Netw.*, Porto, Protugal: Springer, 2007, pp. 291–300.
[43] L. Yin, Z. Deng, B. Huo, Y. Xia, and C. Li, "Robust derivative unscented Kalman filter under non-Gaussian noise," *IEEE Access*, vol. 6, pp. 33129–33136, 2018.



YANG XU received the B.S. degree from Dalian Maritime University, Dalian, China, in 2012, and the M.S. degree from the Department of Aeronautics Engineering, Air Force Engineering University, Xi'an, China, in 2015, where he is currently pursuing the Ph.D. degree.

He has published more than ten articles in journal and conference proceedings, including journals, such as *Electron Optics*, *Infrared and Laser Engineering*, and *Journal of Systems Engineering and Electronics*. His main research interests include infrared confrontation, aircraft guidance law design, and deep learning.

YANGWANG FANG received the B.S. degree from Huaibei Normal University, Huaibei, in 1987, the M.S. degree from Shaanxi Normal University, Xi'an, China, in 1990, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, in 1998.

He is currently a Professor with the Department of Aeronautics and Astronautics Engineering, Air Force Engineering University. He has published more than 200 articles in journal, book, and conference proceedings, including journals, such as the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, Aerospace Science and Technology, Neurocomputing, *IET Radar, Sonar & Navigation*, and *International Journal for Light and Electron Optics*. He has published ten books. His main research interests include guidance and control of the missile, stochastic optimal control theory, and application and nonlinear control.

WEISHI PENG received the B.S. and M.S. degrees from People Armed Police Engineering University, Xi'an, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Aeronautics and Astronautics Engineering, Air Force Engineering University, Xi'an, in 2016.

From 2013 to 2015, he was a Teaching Assistant with the Department of Equipment Management and Support, People Armed Police Engineering University. From 2016 to 2018, he held a postdoctoral position with the School of Aeronautical Engineering, Air Force Engineering University. He was also a Lecturer with the Department of Equipment Management and Support, People Armed Police Engineering University, Xi'an, where he has been an Assistant Professor, since 2019. He is the author of two books, more than 40 articles, and more than 20 inventions. His main research interests include performance evaluation theory, error spectrum theory, swarm intelligence algorithm, and weapon accuracy analysis and evaluation.

YOULI WU received the B.S., M.S., and Ph.D. degrees from the Department of Aeronautics Engineering, Air Force Engineering University, Xi'an, China, in 2002, 2005, and 2009, respectively.

He is currently an Associate Professor with the Department of Aeronautics Engineering, Air Force Engineering University. He has published more than 100 articles in journal, book, and conference proceedings, including journals, such as the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *Aerospace Science and Technology*, and *Neurocomputing*. He has published five books. His main research interests include infrared confrontation, and accuracy analysis and evaluation of the missile.