

# Clustering Methods for Particle Filters with Gaussian Mixture Models

Sehyun Yun<sup>1</sup>, *The University of Texas at Austin*, Austin, Texas 78712

Renato Zanetti<sup>2</sup>, *The University of Texas at Austin*, Austin, Texas 78712

## Abstract

New clustering methods are proposed to develop novel particle filters with Gaussian mixture models (PFGMM). In the PFGMM, the propagated samples are clustered to recover a Gaussian mixture model (GMM) using a clustering algorithm, which plays a fundamental role in the filter's performance. Two clustering methods are introduced that simultaneously minimize the covariance of each of the GMM components and maximize the likelihood function. Under the scenarios considered in this paper, it is shown through numerical simulation that the PFGMMs with the proposed clustering algorithms lead to better performance than the PFGMM employing the K-means or the expectation-maximization (EM) algorithms as well as the regularized particle filter (RPF) and the Gaussian sum particle filter (GSPF).

## I. INTRODUCTION

Filters estimate the state of a dynamic system from a series of noisy measurements. For linear dynamical systems corrupted by additive Gaussian noise, the Kalman filter provides an optimal solution to the minimum mean-square error estimation problem [1]. In practice, however, the nonlinearity and non-Gaussianity in many systems of practical interest make the Kalman filter inapplicable in its basic form. To deal with nonlinear systems, linear estimators of nonlinear systems such as the extended Kalman filter (EKF) [2] and the unscented Kalman filter (UKF) [3] have been developed under the assumption that all distributions are Gaussian. Furthermore, nonlinear filters such as the Gaussian sum filter (GSF) [4], the particle filter (PF) [5], the Gaussian sum particle filter (GSPF) [6], and the PF with Gaussian mixture models (PFGMM) [7] have been used for highly nonlinear/non-Gaussian problems.

<sup>1</sup> Ph.D. candidate, Department of Aerospace Engineering and Engineering Mechanics

<sup>2</sup> Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics

The PFGMM algorithm recently proposed by [8] has shown that it successfully estimates the states of a highly nonlinear system while other filters such as the GSF, bootstrap PF (BPF), auxiliary PF (APF), and regularized PF (RPF) diverge. In the PFGMM, the propagated samples are clustered to recover a Gaussian mixture model (GMM) representation of the prior probability density function (PDF). Ref. [7] demonstrated that the filter's density converges in probability to the true filter density under the assumption of a perfect clustering scheme, implying the filter performance strongly depends on the performance of a clustering method. Popular clustering algorithms such as the K-means algorithm and expectation-maximization (EM) algorithm for GMMs have been applied to the PFGMM [7], [8].

The first PFGMM algorithm proposed, the Particle Gaussian Mixture Filter [7], utilizes K-means clustering which produces separated clusters for each Gaussian component of the GMM. When K is a fixed preassigned number of clusters, the K-means algorithm minimizes within-cluster-sum-of-squares (WCSS) (i.e. variance) by partitioning the data set into the K clusters. As a result, the covariance of each cluster generated by the K-means algorithm is as small as possible. A small covariance of each Gaussian mixture component is desirable outcome as the GSF is globally optimal only for linear systems and GMM distributions. In the presence of nonlinearities, typically, the smaller the covariance of a component, the smaller the nonlinear effects of the measurement function in the likely realizations of that component, and hence the better the performance of the GSF.

Although it produces components with small covariance, the K-means algorithm does not guarantee an accurate GMM representation of the prior PDF. The EM algorithm for GMMs, produces a better approximation of the prior distribution by maximizing the likelihood function with respect to the clustering parameters, which are the means, covariance matrices, and weights of the components of a GMM. The K-means algorithm is a special case of the EM algorithm for GMMs in which all mixture weights are equal and all covariance matrices have spherical forms. The EM algorithm produces larger covariances than K-means, since it is not a hard clustering algorithm and allows for the components to overlap.

In this work, we propose two novel clustering algorithms to merge the benefits of K-means and EM and apply them to the PFGMM to improve its estimation performance. The two new types of clustering methods simultaneously minimize the covariance for each of the components of a GMM and maximize the likelihood function based on a fuzzy C-means (FCM) clustering algorithm [9]. The FCM algorithm is closely related to the K-means algorithm and EM algorithm

for GMMs. In FCM clustering, the data points can belong to more than one cluster with different membership grades between 0 and 1. These membership grades represent the degree to which data points belong to the different clusters. The standard FCM algorithm, which is also referred to as soft K-Means algorithm, employs a weighting exponent on each fuzzy membership. The weighting exponent is also called the fuzzifier since it determines the level of the fuzziness of clustering. Unlike the approach of the standard FCM algorithm, the FCM algorithm can also be regarded as regularization of the K-means algorithm with a maximum entropy method [10]. Based on the new aspect of the FCM algorithm, Ref. [11] proposed the FCM with regularization by Kullback-Leibler information (KLFCM), which is similar to the EM algorithm for GMMs. Inspired from the standard FCM and KLFCM algorithm, in this paper we propose two types of clustering methods specifically designed to improve the performance of the PFGMM.

The remainder of the paper is organized as follows. Sections II and III describe the PFGMM and the FCM algorithm, respectively. Then, the new clustering algorithms are proposed in Section IV. Section V presents simulation results of the proposed algorithms, followed by some concluding remarks on the new methodologies and the results.

## II. PARTICLE FILTER WITH GAUSSIAN MIXTURE MODELS

In the PFGMM [8], each algorithm's iteration starts from the knowledge of the prior distribution  $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$  which is approximated by  $N$  independent and identically distributed (i.i.d.) samples  $\mathbf{x}_{k-1}^{(i)}$

$$p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) \approx \sum_{i=1}^N \frac{1}{N} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(i)}) \quad (1)$$

where  $k$  indicates the discrete time step,  $\delta(\cdot)$  is the Dirac delta function, and  $\mathbf{Y}_{k-1}$  is the set of all measurement vectors  $\{\mathbf{y}_1, \dots, \mathbf{y}_{k-1}\}$  where  $\mathbf{y}_{k-1}$  is a measurement vector at the time step  $k-1$ . Following the same procedure as particle filters [5], a set of samples at the next time step is generated using the Markov transition kernel  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ . The Markov kernel is defined by the dynamics of a system and the known statistics of the process noise.

The next step of the algorithm is to cluster the particles into Gaussian mixtures using a clustering algorithm such as the K-means algorithm or the EM algorithm for GMMs and the propagated distribution is then expressed as follows:

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) \approx \sum_{j=1}^K \omega_{k|k-1}^{(j)} n(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) \quad (2)$$

where  $K$  is the predetermined number of clusters,  $n(\mathbf{x}|\hat{\mathbf{x}}, P)$  represents the Gaussian PDF with mean  $\hat{\mathbf{x}}$  and covariance  $P$ ; and  $\omega_{k|k-1}^{(j)}$ ,  $\hat{\mathbf{x}}_{k|k-1}^{(j)}$ , and  $P_{k|k-1}^{(j)}$  are the weight, mean, and covariance matrix of the  $i$ th Gaussian component calculated by the K-means or EM clustering algorithm. The K-means and EM algorithm are similar in the sense that they use an iterative refinement approach to find the optimal clusters. The K-means is a hard clustering algorithm, which means each particle is associated uniquely with one cluster, as such it uses only the points in the same cluster to update each component's mean. The EM algorithm performs a soft assignment and approximates the PDF of  $\mathbf{x}_k$  with several Gaussian components having different means, covariance matrices, and weights. The details of the K-means and EM algorithm are explained in [12].

Finally, we incorporate measurement information by updating the means, covariance matrices, and weights of all the components in the same way as the measurement update of the GSF. We then draw  $N$  i.i.d. samples from the updated GMM which is the posterior distribution. These samples are the starting point for the next iteration. The details of the measurement update of the GSF and the algorithm to draw  $N$  i.i.d. samples from a GMM are well known and can be found, for example, in [13]. The flow chart of the PFGMM is described in Fig. 2

### III. FUZZY C-MEANS CLUSTERING

The standard FCM and the KLFCM algorithm are reviewed in this section. In the standard FCM algorithm, the aim of the algorithm is to minimize the objective function which is defined as follows [9]:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C (u_{ij})^m d_{ij} \quad (3)$$

where  $C$  is the preassigned number of clusters and  $m$  is the weighting exponent for the fuzzification of memberships,  $m \geq 1$ .  $u_{ij}$  denotes the membership value of the  $i^{th}$  data sample for the  $j^{th}$  cluster and all membership values have the following constraints:

$$u_{ij} \geq 0, \forall i, j, \quad \sum_{j=1}^C u_{ij} = 1 \quad (4)$$

The dissimilarity function  $d_{ij}$  which is a measure of the Euclidean distance between  $i^{th}$  data sample  $\mathbf{x}^{(i)}$  and the center of the  $j^{th}$  cluster  $\boldsymbol{\mu}_j$  is expressed as follows:

$$d_{ij} = (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) \quad (5)$$

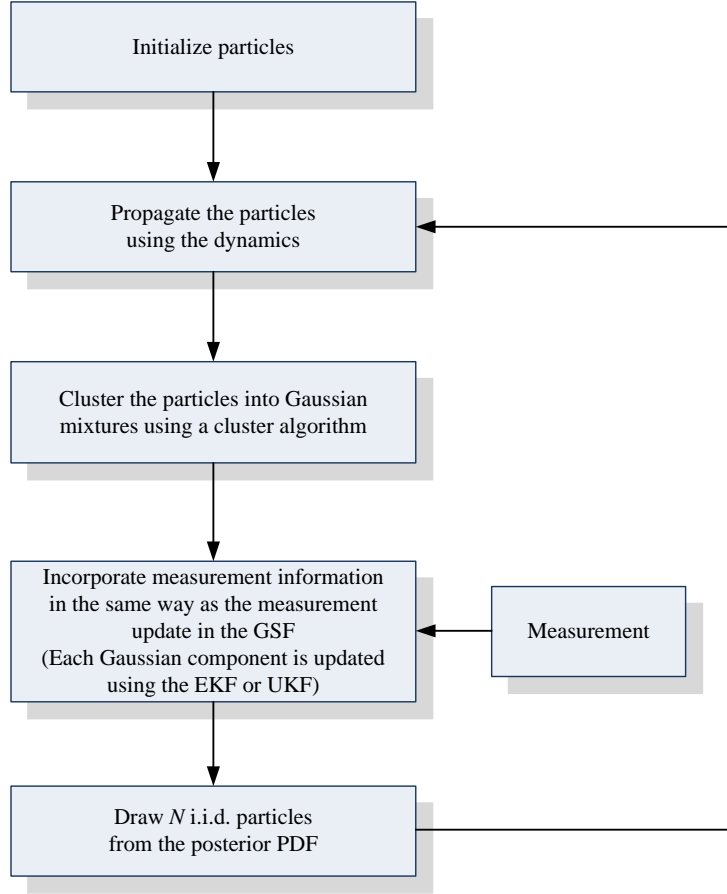


Fig. 1. The flow chart of the PFGMM

The optimal values of the clustering parameters  $u_{ij}$  and  $\mu_j$  are calculated using a fixed-point iteration scheme, which is similar to the EM algorithm for GMMs. The algorithm is summarized as follows:

1. Initialize the membership values  $u_{ij}$  and set the objective function value to infinity.
2. Calculate the cluster centers  $\mu_j$  using the current membership values.

$$\mu_j = \frac{\sum_{i=1}^N (u_{ij})^m \mathbf{x}^{(i)}}{\sum_{i=1}^N (u_{ij})^m} \quad (6)$$

3. Estimate the new membership values  $u_{ij}$  using the current cluster centers to minimize the objective function.

$$u_{ij} = \left[ \sum_{k=1}^C \left( \frac{d_{ij}}{d_{ik}} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (7)$$

4. Evaluate the objective function and check for convergence of it. If the convergence criterion is not satisfied, replace the old membership values with the new ones and return to step 2.

After a number of iterations, the clustering parameters are optimized to minimize the objective function.

Fuzziness is the level of overlap between clusters (more overlap equals less defined or fuzzier boundaries). In the standard FCM algorithm the weighting exponent  $m$  determines the level of the fuzziness of clustering, so it is also called fuzzifier [9]. In other words, the partition is getting fuzzier as the fuzzifier has a larger value. On the other hand, the memberships  $u_{ij}$  converges to 0 or 1 when the fuzzifier  $m$  has the minimum value of one, which means the FCM algorithm reduces exactly to the K-means algorithm.

In addition, the FCM algorithm is closely related to the EM algorithm for GMMs. Ref. [14] defined a fuzzy covariance matrix for the FCM algorithm so that different clusters can have different geometric shapes in the clustering. Moreover, Ref. [10] shows that the FCM algorithm can be regarded as the regularization of the K-means algorithm with a maximum entropy method, and Ref. [11] demonstrates that the EM algorithm of the GMMs can be casted as a penalized version of the hard means clustering algorithm. As a result, the FCM clustering with the regularizer by Kullback-Leibler (KL) information, called KLFCM, can be made the same algorithm as the EM algorithm for GMMs [15]. The objective function of the KLFCM is expressed as follows:

$$J_{KL} = \sum_{i=1}^N \sum_{j=1}^C u_{ij} d'_{ij} + \lambda \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log \frac{u_{ij}}{\pi_j} + \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log |\Sigma_j| - \sum_{i=1}^N \eta_i \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] \quad (8)$$

where the dissimilarity function  $d'_{ij}$  of the KLFCM is the Mahalanobis distance,

$$d'_{ij} = (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) \quad (9)$$

and  $\lambda$  is the fuzzifier,  $\lambda \geq 0$ .  $\Sigma_j$  and  $\pi_j$  are the covariance matrix and weight of the  $j^{th}$  cluster, and  $\eta_i$  and  $\tau$  are Lagrangian multipliers whose corresponding terms respectively indicate the constraints on membership values and weights.

The necessary conditions for optimality of (8) are derived as follows:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^N u_{ij} \mathbf{x}^{(i)}}{\sum_{i=1}^N u_{ij}} \quad (10)$$

$$\Sigma_j = \frac{\sum_{i=1}^N u_{ij} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^N u_{ij}} \quad (11)$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^N u_{ij} \quad (12)$$

where

$$u_{ij} = \frac{\pi_j \exp\left(-\frac{1}{\lambda} d'_{ij}\right) |\Sigma_j|^{-1/\lambda}}{\sum_{k=1}^C \pi_k \exp\left(-\frac{1}{\lambda} d'_{ik}\right) |\Sigma_k|^{-1/\lambda}} \quad (13)$$

The iteration rule is equivalent to the standard FCM algorithm. In the KLFCM algorithm, the fuzzifier  $\lambda$  tunes the degree of fuzziness of the membership values. For instance, all memberships are getting closer to  $1/C$  as we are putting more weight on the relative entropy term. Thus, the larger the fuzzifier, the fuzzier the memberships. The KLFCM algorithm is the same as the EM algorithm for GMMs when the fuzzifier  $\lambda$  is equal to 2.

#### IV. CLUSTERING METHODS FOR PFGMM

This section presents two different clustering methods to simultaneously minimize the covariance for each of the components of a GMM and maximize the likelihood function for the PFGMM. Both these features are key contributors to PFGMM estimation performance.

##### A. KLFCM with Weighting Exponent

The objective function of the KLFCM with the fuzzification coefficient 2 is expressed as follows:

$$\begin{aligned} J_{KL2} = & \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log u_{ij} \\ & + \sum_{i=1}^N \sum_{j=1}^C u_{ij} \left[ \log \frac{1}{\pi_j} - \log \frac{(2\pi)^{-d_x/2}}{(2\pi)^{-d_x/2} |\Sigma_j|^{-1/2}} \exp\left(-\frac{1}{2} d'_{ij}\right) \right] \\ & - \sum_{i=1}^N \boldsymbol{\eta}_i \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] \end{aligned} \quad (14)$$

which is equivalent to the following function:

$$J_{KL2} = \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log u_{ij} + \sum_{i=1}^N \sum_{j=1}^C u_{ij} d''_{ij} - \sum_{i=1}^N \boldsymbol{\eta}_i \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] \quad (15)$$

where  $d_x$  is the dimension of the state  $\boldsymbol{x}$  and the dissimilarity function  $d''_{ij}$  is defined as follows:

$$d''_{ij} = \log \left( \frac{1}{(2\pi)^{-d_x/2} \pi_j n(\boldsymbol{x}^{(i)} | \boldsymbol{\mu}_j, \Sigma_j)} \right) \quad (16)$$

The first term (i.e., entropy term) of the objective function does not have the fuzzifier  $\lambda$  because it is set to be equal to 2. Therefore, the modified KLFCM algorithm with the objective function

could only act as the EM algorithm. To assign the hard clustering property of the K-means to the modified algorithm, we make it include a new fuzzifier. If the new fuzzifier  $\lambda'$  is added to the first term, it has the same form as the FCM algorithm with a maximum entropy method, and the updating rule of the membership values are expressed as follows [10]:

$$u_{ij} = \frac{\exp\left(-\frac{1}{\lambda'} d''_{ij}\right)}{\sum_{k=1}^C \exp\left(-\frac{1}{\lambda'} d''_{ik}\right)} = \frac{\pi_j^{-1/\lambda'} \exp\left(-\frac{1}{2\lambda'} d''_{ij}\right) |\Sigma_j|^{-1/2\lambda'}}{\sum_{k=1}^C \pi_k^{-1/\lambda'} \exp\left(-\frac{1}{2\lambda'} d''_{ik}\right) |\Sigma_k|^{-1/2\lambda'}} \quad (17)$$

Although the memberships become less fuzzier as the fuzzifier  $\lambda'$  is smaller, it simultaneously loses the property of the EM algorithm since the fuzzifier changes and distorts the Gaussian distributions. Consequently, the new fuzzifier makes the modified algorithm very similar to the KLFCM algorithm. To cope with the problem, we can change the objective function into the standard FCM form by removing the first term and introducing the weighting exponent ( $m \geq 1$ ) on the membership values. Then, the modified objective function is expressed as follows:

$$J_{KL2'} = \sum_{i=1}^N \sum_{j=1}^C (u_{ij})^m d''_{ij} - \sum_{i=1}^N \eta \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] \quad (18)$$

The updating rules for the means, covariance matrices, weights of all clusters and memberships are then derived as follows:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^N (u_{ij})^m \mathbf{x}^{(i)}}{\sum_{i=1}^N (u_{ij})^m} \quad (19)$$

$$\Sigma_j = \frac{\sum_{i=1}^N (u_{ij})^m (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^N (u_{ij})^m} \quad (20)$$

$$\pi_j = \frac{1}{N} \sum_{i=1}^N (u_{ij})^m \quad (21)$$

and

$$u_{ij} = \left[ \sum_{k=1}^C \left( \frac{d''_{ij}}{d''_{ik}} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (22)$$

The proposed algorithm maximizes the log likelihood function, which is the same as the EM algorithm for GMMs. Moreover, the memberships are proportional to the exponential of the log likelihood function of the GMM. In other words, even if the weighting exponent  $m$  is changed to tune the level of the fuzziness of the clustering, the form of the log likelihood function (i.e., GMM) in the memberships is maintained. Therefore, the degree of fuzziness of the memberships can be determined by the weighting exponent  $m$  with retaining the property of the EM algorithm. As in the standard FCM algorithm, the partition becomes more distinct as the fuzzifier  $m$  has



a smaller value. When  $m$  is set to be very close to the minimum value of 1, however, most of the memberships converges to 0 or 1, thus making some weights of the clusters zero [9]. As a result, the covariance for the rest of the components of a GMM is increased. To prevent this problem, a regularization term with a positive parameter  $\kappa$  is included in the objective function as follows:

$$J_{mKL} = \sum_{i=1}^N \sum_{j=1}^C (u_{ij})^m d_{ij}'' - \kappa \sum_{i=1}^N \sum_{j=1}^C \log(\pi_j) - \sum_{i=1}^N \eta \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] \quad (23)$$

By introducing this new regularization term, all the weights of the clusters are getting closer to  $1/C$  as the value of  $\kappa$  is larger because the sum of  $\pi_j$  with respect to  $C$  is constrained to be 1. The updating rule for the weights of the clusters is then changed into

$$\pi_j = \frac{2 \sum_{i=1}^N (u_{ij})^m + \kappa N}{2 \sum_{i=1}^N \sum_{j=1}^C (u_{ij})^m + \kappa NC} \quad (24)$$

and the others (19), (20), and (22) remain the same. The proposed algorithm is named the KLFCM clustering with weighting exponent (mKLFCM).

### B. KLFCM with Ridge Regularization

Another proposed clustering method is to add a new regularization term to the objective function of the KLFCM with the fuzzification coefficient 2. Regularization is one of the most important concepts in machine learning (ML) and the most well known regularization techniques are ridge and least absolute shrinkage and selection operator (LASSO) regularization: the two regularization techniques are used to reduce the magnitude of irrelevant coefficients of a model and avoid overfitting [16]. As done in ridge regularization, a new regularization term is added to (8) with the fuzzification coefficient 2 in order to restrict the determinant of the covariance of each cluster to the determinant of the total covariance divided by the number of total clusters as follows:

$$J_{RKL} = \sum_{i=1}^N \sum_{j=1}^C u_{ij} d_{ij}' + \alpha \sum_{i=1}^N \sum_{j=1}^C \log \left( |\Sigma_j| - \frac{|\Sigma_T|}{C} \right)^2 + 2 \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log \frac{u_{ij}}{\pi_j} \quad (25)$$

$$+ \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log |\Sigma_j| - \sum_{i=1}^N \eta_i \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right]$$

where  $\alpha$  is the penalty parameter and  $\Sigma_T$  is the covariance matrix of the total samples. To put it another way, the new clustering algorithm minimizes the following objective function:

$$J_{KL2} \quad \text{subject to} \quad \sum_{j=1}^C \log \left( |\Sigma_j| - \frac{|\Sigma_T|}{C} \right)^2 < R_c \quad (26)$$

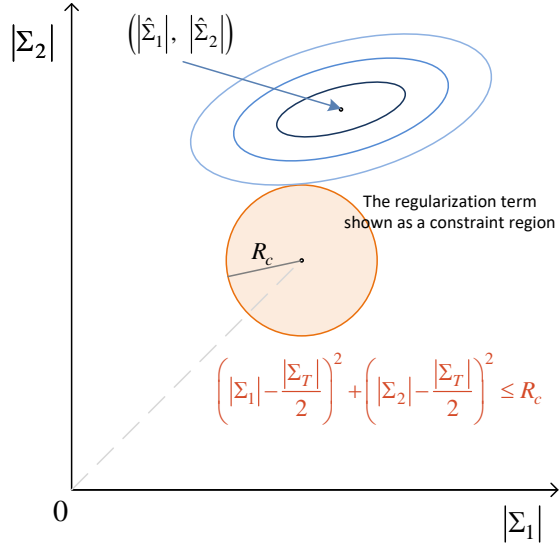


Fig. 2. Geometric interpretation of the new regularizer

where  $J_{KL2}$  is given by (15) and  $R_c$  is a parameter. Fig. 2 shows an example of how the new regularization term in the objective function can be geometrically interpreted when the number of clusters is 2. If the new regularization term does not exist (i.e.  $\alpha = 0$ ), which means there is no inequality constraint, the center of the ellipse will be the optimal value of the determinant of the each cluster's covariance. Fig. 2 shows, however, the optimal values for the covariance matrices of the objective functions in (25) and (26) are given by the first point at which the ellipse (blue) contacts the constraint region (orange) due to the new regularizer (the inequality constraint). As a result, the large value of  $\alpha$  (the smaller value of  $R_c$ ) will shrink the determinants of all the covariance matrices towards the same value ( $|\Sigma_T|/C$ ).

The updating rules for the clustering parameters of (25) are the same as those of (8) with  $\lambda = 2$  except for the covariance matrices. For the new objective function (25), the updating rule for the covariance matrices is not fixed because it varies depending on the dimension of state. Therefore, to find the consistent closed-form solution for the updating rule of the covariance matrices, we substitute  $\Sigma_j$  with  $\beta_j M_j$  where  $|M_j| = 1$ ,  $\beta_j = |\Sigma_j|^{1/d_x}$ . Then, the objective function (25) can

be changed into the following equivalent objective function:

$$\begin{aligned}
J_{RKL} = & \sum_{i=1}^N \sum_{j=1}^C u_{ij} d'_{ij} + \alpha \sum_{i=1}^N \sum_{j=1}^C \left( \frac{D}{\beta_j} - 1 \right)^2 + 2 \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log \frac{u_{ij}}{\pi_j} \\
& + \sum_{i=1}^N \sum_{j=1}^C u_{ij} \log |\Sigma_j| - \sum_{i=1}^N \boldsymbol{\eta}_i \left[ \sum_{j=1}^C u_{ij} - 1 \right] - \tau \left[ \sum_{j=1}^C \pi_j - 1 \right] - \sum_{j=1}^C \xi_j \log |M_j|
\end{aligned} \tag{27}$$

where  $\xi_1, \dots, \xi_C$  are Lagrangian multipliers and

$$D = \left( \frac{|\Sigma_T|}{C} \right)^{1/d_x} \tag{28}$$

Then, the updating rules for the  $M_j$  and  $\beta_j$  are derived as follows:

$$M_j = \frac{\sum_{i=1}^N u_{ij} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T}{\left| \sum_{i=1}^N u_{ij} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T \right|^{1/d_x}} \tag{29}$$

$$\beta_j = \frac{1}{2d_x \sum_{i=1}^N u_{ij}} \left[ A - B + \sqrt{(A - B)^2 + C} \right] \tag{30}$$

where

$$A = \sum_{i=1}^N u_{ij} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T M_j^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) \tag{31}$$

$$B = 2\alpha ND \tag{32}$$

and

$$C = 8\alpha d_x N D^2 \sum_{i=1}^N u_{ij} \tag{33}$$

The proposed algorithm is named the KLFCM clustering with ridge regularization (RKLFCM).

### C. Analysis of the Proposed Clustering Methods

In the mKLFCM clustering algorithm, it is crucial to select the most appropriate weighting exponent  $m$  which determines the performance of the PFGMM. It is desirable for the PFGMM to have an accurate GMM representation of the prior PDF while each component has a small enough covariance matrix such that nonlinear measurements can be accurately approximated by linearization in the support of each component. In this paper, we investigate the impact of the weighting exponent on the clustering algorithm and find the optimal value for the PFGMM based on the biggest determinant of the covariance matrix of the components and the Jensen-Shannon distance (JSD). Note that the smaller the biggest determinant of the covariance matrix, the better

the performance of the PFGMM. The JSD, which is a metric, is used to measure the similarity between two probability distributions and given by

$$J(p_x||q_x) = \sqrt{\frac{1}{2}D_{KL}(p_x||m_x) + \frac{1}{2}D_{KL}(q_x||m_x)} \quad (34)$$

where

$$D_{KL}(p_x||q_x) = \sum_{\mathbf{x} \in S} p_x(\mathbf{x}) \log \left( \frac{p_x(\mathbf{x})}{q_x(\mathbf{x})} \right) \quad (35)$$

$$m_x = \frac{1}{2}(p_x + q_x) \quad (36)$$

and  $S$  is the support for  $\mathbf{x}$ , and  $p_x$  and  $q_x$  are two different probability distributions. As the difference between the two distributions becomes smaller, the JSD also gets smaller.

To analyze the impact of the weighting exponent on the clustering algorithm, a Monte Carlo analysis is performed with 100 simulations. For each simulation, 10,000 particles are drawn from the two-dimensional normal distribution. The mKLFCM uses 3 clusters and the parameter  $\kappa$  is set to be 0.05. Fig. 3 shows the normalized maximum value of determinant and JSD according to the weighting exponent and the original values are listed in Table I. As mentioned above, the greater the weighting exponent is, the lower the JSD value becomes. However, the maximum value of the determinant of covariance also tends to increase with the weighting exponent. Fig. 3 shows that the optimal value of the weighting exponent is 1.216 when samples are from a normal distribution. In addition, Table I implies that as the weighting exponent is larger, the maximum value of determinant approaches to 1, which is the determinant of the sample covariance. Overall, the mKLFCM behaves like the K-means algorithm as the weighting exponent decreases, whereas it functions like the EM algorithm as the weighting exponent increases.

A key element of the RKLFCM is how to determine the penalty parameter  $\alpha$ . A Monte Carlo analysis with the same simulation conditions as above is performed. Fig. 4 presents the

TABLE I  
ANALYSIS OF THE WEIGHING EXPONENT IN THE mKLFCM

	1.05	1.10	1.15	1.20	1.25	1.30	1.35
Maximum determinant	0.2302	0.2510	0.2848	0.3690	0.5686	0.9484	0.9737
JSD	1.8440	1.6716	1.4339	0.9991	0.4598	0.1415	0.1318

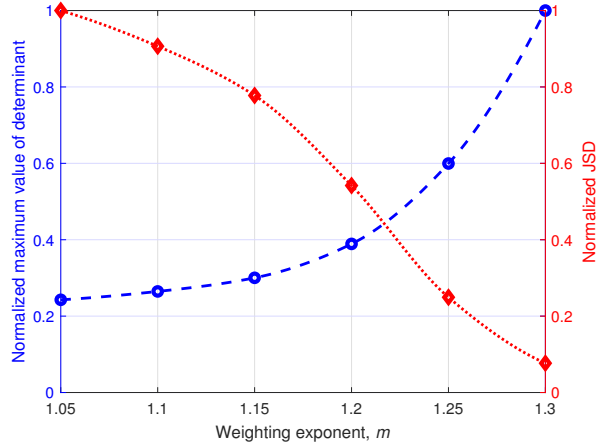


Fig. 3. Analysis of the weighing exponent in the mKLFCM

normalized maximum value of determinant and JSD according to the penalty parameter and the original values are illustrated in Table II. As the penalty parameter increases, the maximum value of determinant decreases. Table II shows that the maximum value of determinant approaches to  $1/3$  which is ratio of the sample covariance to the number of clusters used for the algorithm. On the other hand, the JSD value tends to increase with the penalty parameter. As can be seen in Fig. 4, 0.1414 is the optimal value of the parameter when samples are drawn from a normal distribution. To summarize: the RKLFCM becomes similar to the EM algorithm as the weighting exponent decreases, whereas it operates in a similar way of the K-means algorithm as the weighting exponent increases.

## V. NUMERICAL RESULTS

To evaluate the clustering algorithms proposed in this paper, three different examples are considered: a simple motivating example (used in Refs. [17], [18]), the vector nonstationary

TABLE II  
ANALYSIS OF THE PENALTY PARAMETER IN THE RKLFCM

	0.0001	0.001	0.01	0.1	1	10	100
Maximum determinant	0.7774	0.7714	0.7179	0.5032	0.3567	0.3355	0.3334
JSD	0.2338	0.2365	0.2654	0.4917	0.8018	0.8691	0.8766

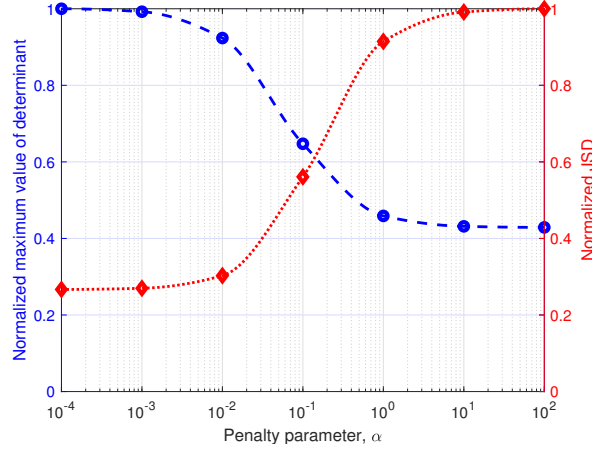


Fig. 4. Analysis of the penalty parameter in the RKLFCM

growth model (used in Refs. [19], [20]), and a Lorenz96 system (used in Refs. [7], [21]).

### A. Single Step Example

Consider the following simple motivating example. An initial bivariate normal random vector  $\mathbf{x}_0$  is distributed as

$$\mathbf{x}_0 \sim n \left( \mathbf{x}_0; \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & -0.1 \\ -0.1 & 0.4 \end{bmatrix} \right) \quad (37)$$

and a range measurement of 1.5 with measurement noise variance of 0.01 is available. Fig. 5 shows contour plots of the given prior and likelihood distribution, and Fig. 6 presents a contour plot of the true posterior distribution. The measurement is nonlinear so that it causes the banana-shaped posterior distribution which is cannot be accurately estimated by linear estimators such as EKF and UKF [17], [18].

In this example, we compare the clustering performances of the four different clustering methods; the K-means algorithm, the EM algorithm for GMMs, and the two algorithms proposed here [mKLFCM and RKLFCM]. We test the clustering algorithms to see whether they approximate (a) the prior distribution by a GMM, (b) the nonlinear measurement function by a linear function for the measurement update of each component, and (c) the true posterior distribution by an updated GMM.

For Case (a) and (c), the JSD is used to measure the distance between the true distribution and approximated distribution by a GMM. Moreover, for each Gaussian component, the approximated

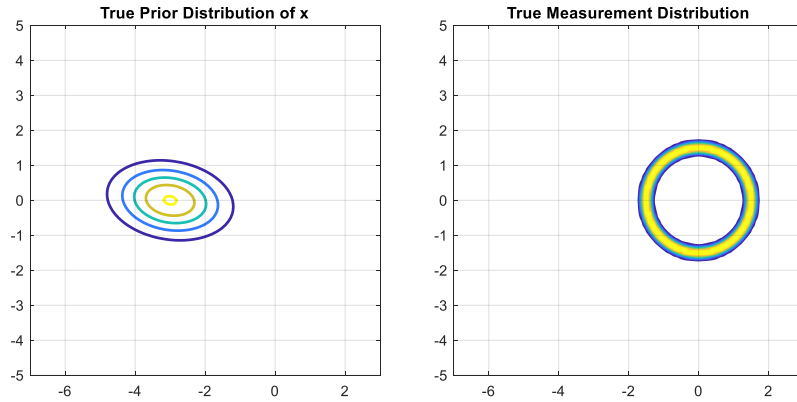


Fig. 5. Prior and measurement distribution for example 1

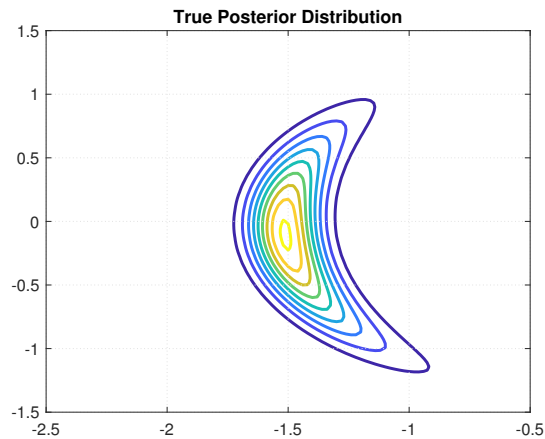


Fig. 6. True posterior distribution for example 1

posterior distribution of the EKF solution is compared to that of the second-order EKF (SEKF) solution using the JSD for Case (b) in order to check whether the linearization of the measurement function is valid in a region of the component [17]. If the covariance of a component is small enough such that the nonlinearities of the measurement function are negligible, then the difference between the distributions will be small. For Case (b), the weighted sum of the JSD values is used for a GMM with the weights of all the components.

All of the clustering algorithms use 30 clusters with 10,000 particles. The proposed clustering algorithms use the following tuning parameters:  $m = 1.216$  and  $\kappa = 0.05$  for the mKLFCM, and  $\alpha = 0.1414$  for the RKLFCM. For the EM, mKLFCM, and RKLFCM algorithm, the

TABLE III  
MONTE CARLO AVERAGED JSD FOR EXEMPLE 1 (100 SIMULATIONS)

	K-means	EM	mKLFCM	RKLFCM
Case (a)	2.6467	0.5866	1.5423	0.7044
Case (b)	2.3689	5.8114	3.4402	4.5364
Case (c)	6.2189	6.0616	5.2056	5.0687

initial clustering parameters are selected using the K-means algorithm [12]. A Monte Carlo analysis is performed with 100 simulations and the averaged JSD values for all of the cases are listed in Table III. For Case (a), the EM algorithm for GMMs and RKLFCM algorithm are comparable, while the K-Means algorithm results in the worst performance among the four clustering algorithms. For Case (b), however, the K-means algorithm is better than any of other algorithms and the EM algorithm gives the worst performance. Finally, Table III shows that, for Case (c), the proposed algorithms [mKLFCM and RKLFCM] outperform the K-means and EM algorithm for GMMs, which means the proposed algorithms are more suitable for the PFGMM.

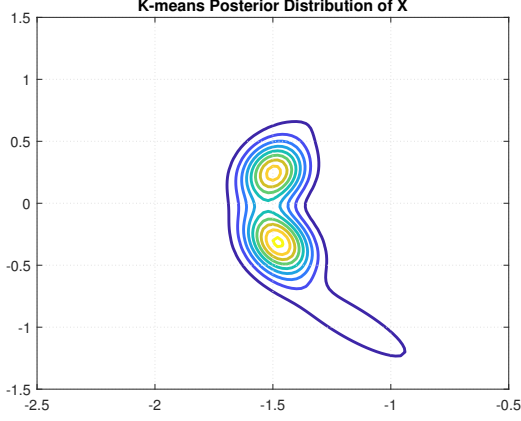
Moreover, for Case (c), Fig. 7 displays the PFGMM solutions with the K-means, EM for GMMs, mKLFCM, and RKLFCM clustering algorithms for a single simulation. Note that the PFGMM solutions with the four clustering methods succeed in capturing the curvature shape of the true posterior distribution. If a linear estimator of nonlinear systems such as the EKF or UKF is used for this example, the curvature shape of the true distribution cannot be replicated by the first two moments of the estimator. This figure also shows that the PFGMM solutions with the proposed clustering methods can more accurately adapt to nonlinearities of the measurement function than the PFGMM with the K-means and EM algorithm for GMMs.

### B. Vector Nonstationary Growth Model

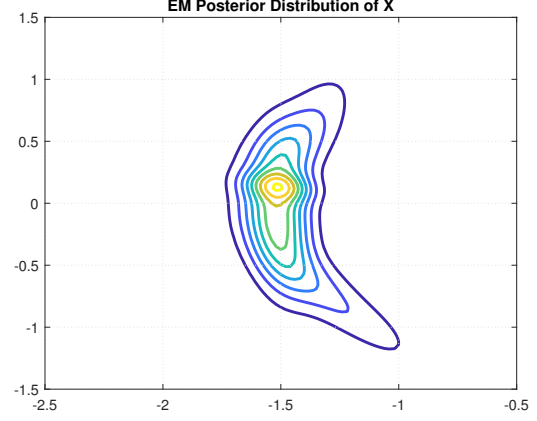
Consider the discrete-time highly nonlinear bivariate dynamic system and measurement model given by [19], [20]

$$\begin{aligned}
 x_{k+1,1} &= \frac{x_{k,1}}{2} + 25 \frac{x_{k,1}}{1 + x_{k,1}^2} + 8 \cos(1.2k) + \nu_{k,1} \\
 x_{k+1,2} &= \frac{x_{k,2}}{2} + 25 \frac{x_{k,2}}{1 + x_{k,2}^2} + 8 \cos(1.2k) + \nu_{k,2}
 \end{aligned} \tag{38}$$

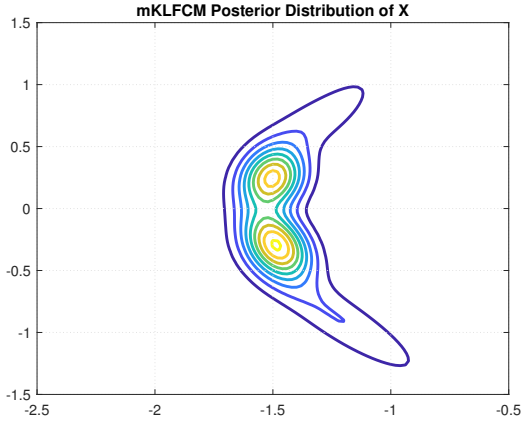




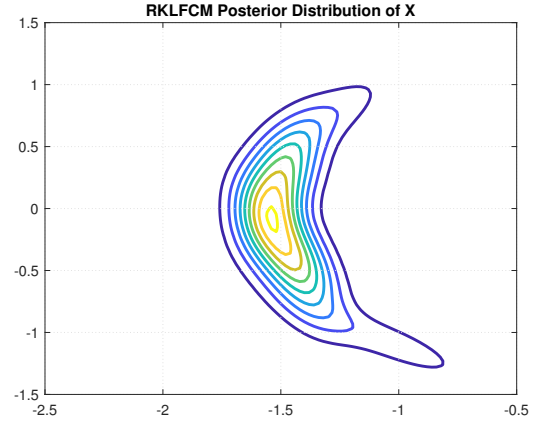
(a) PFGMM with the K-means algorithm Posterior



(b) PFGMM with the EM algorithm Posterior



(c) PFGMM with the mKLFCM algorithm Posterior



(d) PFGMM with the RKLFCM algorithm Posterior

Fig. 7. The PFGMM solutions for Example 1 are shown with the four different clustering algorithms: (a) the JSD value (the K-means) = 5.6130, (b) the JSD value (the EM algorithm for GMMs) = 5.4898, (c) the JSD value (the mKLFCM) = 4.4212, and (d) the JSD value (the RKLFCM) = 4.2193.

$$\begin{aligned}
 y_{k+1,1} &= \frac{x_{k+1,1}^2 + x_{k+1,2}^2}{20} + \eta_{k+1,1} \\
 y_{k+1,2} &= \frac{x_{k+1,1}^2 - x_{k+1,2}^2}{10} + \eta_{k+1,2}
 \end{aligned} \tag{39}$$

where the process noise  $\boldsymbol{\nu}_k = [\nu_{k,1}, \nu_{k,2}]^T$  and the measurement noise  $\boldsymbol{\eta}_{k+1} = [\eta_{k+1,1}, \eta_{k+1,2}]^T$  are assumed to be independent zero-mean Gaussian random variables with variances of  $Q = 10I_{2 \times 2}$  and  $R = I_{2 \times 2}$ , respectively, and  $I_{2 \times 2}$  is a  $2 \times 2$  identity matrix.

The model is highly nonlinear and the cosine term in the dynamic equation varies with time  $k$ . In this example, a Monte Carlo analysis is performed with 200 simulations, and each simulation

has a time span of  $k = [0, 100]$ . The estimation performances of the PFGMM with the four different clustering algorithms, the RPF, and the GSPF are compared based on the root-mean-square error (RMSE) and the noncredibility index (NCI) [22]. The RMSE for each Monte Carlo simulation is computed from the true and estimated states at each time  $k$ . The NCI metric is defined as

$$\text{NCI}_k = \frac{1}{M} \sum_{j=1}^M [10 \log_{10} ((\mathbf{x}_k^j - \boldsymbol{\mu}_k^j)^T (\mathbf{P}_k^j)^{-1} (\mathbf{x}_k^j - \boldsymbol{\mu}_k^j)) - 10 \log_{10} ((\mathbf{x}_k^j - \boldsymbol{\mu}_k^j)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_k^j - \boldsymbol{\mu}_k^j))] \quad (40)$$

where  $M$  is the number of Monte Carlo simulations,  $\mathbf{x}_k^j$  are the true states,  $\boldsymbol{\mu}_k^j$  are the estimated states,  $\mathbf{P}_k^j$  are the filter's error covariance matrix of the  $j$ -th Monte Carlo run, and  $\boldsymbol{\Sigma}_k$  is the ensemble error covariance matrix of the estimates at time  $k$  computed from the Monte Carlo samples. The normalized estimation error squared (NEES) [23] is often used for a standard consistency metric, however, it penalizes optimism much more than pessimism. On the other hand, the NCI metric is a balanced measure of the consistency of the estimators, i.e., the NCI metric is a geometric average of 10 times the logarithm of the NEES ratio [22]. Therefore, the NCI measures the difference between the ideal error covariance matrix  $\boldsymbol{\Sigma}_k$  and the estimated error covariance matrix  $\mathbf{P}_k$ . When the difference between the ensemble error covariance matrix and the filter's error covariance matrix is small, the NCI value should be zero or nearly zero at all times. A total of 200 particles is used in all of the algorithms and the UKF measurement update for 3 clusters is used for the PFGMM. In the GSPF, 3 Gaussian components are employed and the importance sampling function is obtained from the measurement update step of the UKF. The tuning parameters for the proposed algorithms and selecting method for initial clustering parameters are the same as those used in the example 1.

Fig. 8 shows the RMSE and the absolute NCI value of the 200 simulations. The time-averaged value of the RMSE and NCI of the six algorithms are described in Table IV. The proposed algorithms outperform the K-means and EM algorithm for GMMs in terms of accuracy and consistency. Moreover, the figure shows that the PFGMMs perform better than the RPF and GSPF both in the accuracy and consistency. The computation time for filtering run in MATLAB on a 3.2 GHz hexa-core Windows operation systems is also presented in Table IV. In terms of computation time, the proposed algorithms are cheaper than the EM algorithm for GMMs and RPF while the best performance is produced with the GSPF algorithm.

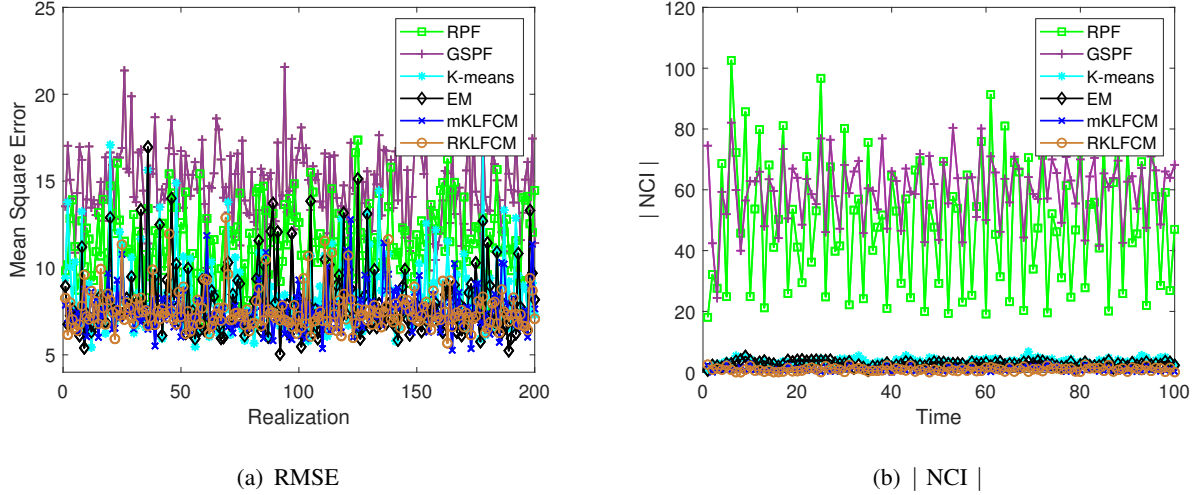


Fig. 8. The average RMSE and the absolute NCI value for 200 Monte Carlo simulations for Example 2

TABLE IV  
MONTE CARLO AVERAGED RMSE, NCI, AND COMPUTATION TIME FOR EXAMPLE 2 (200 SIMULATIONS)

	RMSE	NCI	Computation time
RPF	11.7031	49.4944	1.4342
GSPF	12.4174	60.0373	0.2075
K-Means	8.7378	3.6591	0.4889
EM	8.2914	2.7653	1.0130
mKLFCM	7.6699	1.0270	0.7232
RKLFCM	7.6417	0.9482	0.6525

### C. Lorenz96 System

In this scenario, the RPF, the GSPF, and the PFGMM with the four different clustering algorithms are applied to a Lorenz96 system [7], [21]. The dynamics of the Lorenz96 system is expressed as follows:

$$\dot{x}_i(t) = x_{i-1}(t) \left( x_{i+1}(t) - x_{i-2}(t) \right) - x_i(t) + F + \nu_i(t) \quad (41)$$

$$\mathbf{y}_k = H \mathbf{X}(t_k) + \boldsymbol{\eta}_k, \quad H_{i,j} = \begin{cases} 1, & j = 2i - 1 \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } i = 1, \dots, 20, j = 1, \dots, 40 \quad (42)$$

where  $x_i(t)$ ,  $i = 1, 2, \dots, 40$ , represents the components of the 40th-dimensional vector  $\mathbf{X}(t)$ . In the dynamics equation, the following conventions are used:  $x_{-1} = x_{N-1}$ ,  $x_0 = x_N$ , and

TABLE V  
MONTE CARLO AVERAGED RMSE, NCI, AND COMPUTATION TIME FOR EXAMPLE 3 (100 SIMULATIONS)

	RMSE	NCI	Computation time
RPF	21.7334	7.0335	4.6449
GSPF	23.3898	30.9887	2.2909
K-Means	19.2611	0.3896	2.1983
EM	19.0662	0.4747	3.0462
mKLFCM	18.9772	0.3343	2.3537
RKLFCM	18.6428	0.4683	2.8197

$x_1 = x_{N+1}$ . The constant external force,  $F$ , is selected equal to 8 to cause chaotic behavior in the system. The dynamics is propagated for 10 seconds at 20 Hz,  $t_k = 1, 2, \dots, 200$ , and fourth order Runge-Kutta integration is implemented with a step size of 0.05 second. The process noise,  $\nu_i(t)$ , remains constant over each 0.05 second interval without correlation between the intervals. The discrete linear measurements are available at 1 Hz and measure only the components of the state vector that have odd indices. It is assumed that the process noise and measurement noise,  $\eta_k$ , are uncorrelated, white, zero mean, and with covariance matrices given by  $Q = 10^{-2}$  and  $R = 10^{-2}\mathbf{I}_{20 \times 20}$ , respectively. The initial distribution of the system is assumed to be Gaussian distribution with  $\boldsymbol{\mu}_0 = F[1, 1, \dots, 1]^T$  and  $P_0 = 10^{-3}\mathbf{I}_{40 \times 40}$ .

A Monte Carlo analysis is performed with 100 simulations to evaluate the performance of the RPF, GSPF, and PFGMMs with 2000 particles for the Lorenz96 system. The PFGMMs use 5 clusters and the GSPF uses 5 Gaussian components. The other conditions for all of the algorithms are the same as those used in the example 2. Figure 9 shows the time history of the RMSE and the absolute NCI value of 100 simulations. The results show that the PFGMMs have better performance than the RPF and GSPF in terms of accuracy and consistency. The quantitative results representing the accuracy, consistency, and computation time of the filters are listed in Table V. The proposed algorithms outperform the K-means and EM algorithm for GMMs in terms of accuracy. Moreover, in terms of computation time, the proposed algorithms are faster than the EM algorithm for GMMs and RPF while the best performance is produced with the K-means algorithm.

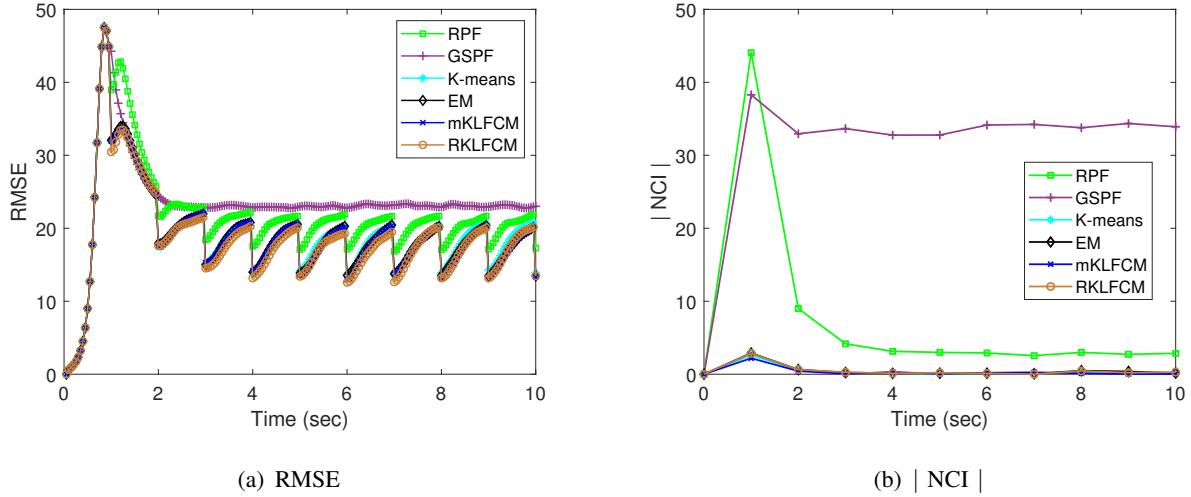


Fig. 9. The average RMSE and the absolute NCI value for 100 Monte Carlo simulations for Example 3

## VI. CONCLUSIONS

In this paper, two new clustering algorithms are proposed whose performance index minimizes the covariance of each of the components of a Gaussian mixture model and maximizes the likelihood function simultaneously. The two new clustering algorithms are based on fuzzy C-means with regularization by Kullback-Leibler information. The objective function of the first method has the same form of the standard fuzzy C-means algorithm with an added weighting exponent. In the second method, a new regularization term is included in the objective function of the baseline algorithm. As a result, both of the proposed clustering algorithms compensate for the drawbacks of the K-means and expectation-maximization algorithm for the particle filter with Gaussian mixture models. Three numerical examples show that the particle filter with Gaussian mixture models with the proposed clustering algorithms provide better performance than the particle filter with Gaussian mixture models with the K-means or expectation-maximization algorithm as well as the regularized particle filter and the Gaussian sum particle filter in terms of the accuracy and consistency.

## ACKNOWLEDGMENT

This work was sponsored in part by the U.S. Air Force Office of Scientific Research under Grant No. FA9550-18-1-0351.

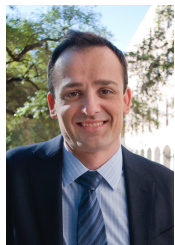
## REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, March 1960, doi:10.1115/1.3662552.
- [2] A. Gelb, Ed., *Applied Optimal Estimation*. Cambridge, MA: The MIT press, 1974.
- [3] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," *SPIE Proceedings*, vol. 3068, 1997, doi:10.1117/12.280797.
- [4] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, August 1972, doi:10.1109/SAP.1970.270017.
- [5] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002, doi:10.1109/78.978374.
- [6] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct 2003, doi:10.1109/TSP.2003.816754.
- [7] D. Raihan and S. Chakravorty, "Particle gaussian mixture filters-i," *Automatica*, vol. 98, pp. 331–340, 2018, doi:10.1016/j.automatica.2018.07.023.
- [8] S. Yun and R. Zanetti, "Nonlinear filtering of light-curve data," *Advances in Space Research*, vol. 66, no. 7, pp. 1672–1688, 2020, doi:10.1016/j.asr.2020.06.024.
- [9] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum Press, 1981.
- [10] M. Sadaaki and M. Masao, "Fuzzy c-means as a regularization and maximum entropy approach," in *7th International Fuzzy Systems Association World Congress*, vol. 2, Jun. 1997, pp. 86–92.
- [11] R. J. Hathaway, "Another interpretation of the em algorithm for mixture distributions," *Statistics and Probability Letters*, vol. 4, no. 2, pp. 53–56, 1986, doi:10.1016/0167-7152(86)90016-7.
- [12] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer-Verlag, 2006.
- [13] S. Yun and R. Zanetti, "Sequential monte carlo filtering with gaussian mixture sampling," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 9, pp. 2069–2077, 2019, doi:10.2514/1.G004403.
- [14] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, Jan. 1979, pp. 761–766, doi:10.1109/CDC.1978.268028.
- [15] K. H. Hidetomo Ichihashi, Kiyotaka Miyagishi, "Fuzzy c-means clustering with regularization by k-l information," in *10th IEEE International Conference on Fuzzy Systems*, Dec. 2001, pp. 924–927, doi:10.1109/FUZZ.2001.1009107.
- [16] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *21st International Conference on Machine Learning*, Jul. 2004, doi:10.1145/1015330.1015435.
- [17] K. Tuggle and R. Zanetti, "Automated splitting gaussian mixture nonlinear measurement update," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 3, pp. 725–734, 2018, doi:10.2514/1.G003109.
- [18] R. Zanetti and K. Tuggle, "A novel gaussian mixture approximation for nonlinear estimation," in *21st International Conference on Information Fusion (FUSION)*, Jul. 2018, doi:10.23919/ICIF.2018.8455485.
- [19] M. Murata, I. Kawano, and K. Inoue, "Extended, unscented kalman, and sigma point multiple distribution estimation filters for nonlinear discrete state-space models," *IEEE Control Systems Letters*, vol. 4, no. 4, pp. 982–987, 2020, doi:10.1109/LCSYS.2020.2998578.
- [20] B. Jia and M. Xin, "Refined nonlinear gaussian quadrature filter," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 5366–5371, doi:10.23919/ACC.2019.8814636.

- [21] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, “A local ensemble kalman filter for atmospheric data assimilation,” *Tellus A: Dynamic Meteorology and Oceanography*, vol. 56, no. 5, pp. 415–428, 2004, doi:10.3402/tellusa.v56i5.14462.
- [22] X. R. Li and Z. Zhao, “Measuring estimator’s credibility: Noncredibility index,” in *9th International Conference on Information Fusion (FUSION)*, Jul. 2006, doi:10.1109/ICIF.2006.301770.
- [23] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2001.



**Sehyun Yun** received the Ph.D. degree in Aerospace Engineering from The University of Texas at Austin, Austin, TX, USA, in May 2021. He is currently a postdoctoral fellow at UT Austin. His research interests include space situational awareness, nonlinear estimation, and hybrid model-/data-driven estimation techniques.



**Renato Zanetti** is an Assistant Professor of Aerospace Engineering at The University of Texas at Austin. Prior to joining UT he worked as an engineer at the NASA Johnson Space Center and at Draper Laboratory. Renato’s research interest include nonlinear estimation, onboard navigation, and autonomous aerospace vehicles.