Chinese Journal of Aeronautics, (2014),27(5): 1242–1250

CSAA

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Adaptive Gaussian sum squared-root cubature Kalman filter with split-merge scheme for state estimation

Liu Yu *, Dong Kai, Wang Haipeng, Liu Jun, He You, Pan Lina

Institute of Information Fusion, Naval Aeronautical and Astronautical University, Yantai 264001, China

Received 23 June 2013; revised 4 December 2013; accepted 6 May 2014
Available online 28 September 2014

KEYWORDS

Adaptive split-merge scheme;
Gaussian sum filter;
Nonlinear non-Gaussian;
State estimation;
Squared-root cubature Kalman filter

Abstract The paper deals with state estimation problem of nonlinear non-Gaussian discrete dynamic systems for improvement of accuracy and consistency. An efficient new algorithm called the adaptive Gaussian-sum square-root cubature Kalman filter (AGSSCKF) with a split-merge scheme is proposed. It is developed based on the squared-root extension of newly introduced cubature Kalman filter (SCKF) and is built within a Gaussian-sum framework. Based on the condition that the probability density functions of process noises and initial state are denoted by a Gaussian sum using optimization method, a bank of SCKF are used as the sub-filters to estimate state of system with the corresponding weights respectively, which is adaptively updated. The new algorithm consists of an adaptive splitting and merging procedure according to a proposed split-decision model based on the nonlinearity degree of measurement. The results of two simulation scenarios (one-dimensional state estimation and bearings-only tracking) show that the proposed filter demonstrates comparable performance to the particle filter with significantly reduced computational cost.

1. Introduction

State estimation of nonlinear discrete-time stochastic dynamic systems is a fast growing area playing a crucial role in many fields such as target tracking[1], satellite navigation, signal processing, fault detection, adaptive and optimal control problems and decision-making process.[2]

A general solution to recursive state estimation problems, based on the Bayesian approach, is given by the Bayesian recursive relation (BRR) for probability density functions (PDFs) of the state conditioned by the measurements. An optimal estimate of the target state is then computed from the posterior density. However, due to the nonlinearities in the system measurement model of the stochastic dynamic systems, the optimal solution from Bayesian filtering is mathematically intractable.[3]

In the last decade, novel approaches to suboptimal filter design based on the polynomial interpolation or the unscented transformation, have been published.

* Corresponding author. Tel.: +86 535 6635877.
E-mail address: liuyu77360132@126.com (Y. Liu).
Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.cja.2014.09.007

The suboptimal methods approximate nonlinear functions in the state or the measurement equation by the Taylor series up to the first or second order. The BRR's solution based on this approximation leads to e.g. the extended Kalman filter or the second order filter,[4] which linearizes the measurement model and often results in unstable performances, including poor estimate accuracy and divergences.[5]

The approximation of the nonlinear functions by means of Stirling's polynomial interpolation leads to the divided difference filters.[6] Instead of a direct replacement of the nonlinear functions in the system description by their approximation, some moment-matching filters such as the unscented Kalman filter (UKF),[7] the Gauss-Hermite filter[8] and the cubature Kalman Filter[9] deterministically select a set of weighted sample points to approximate the posterior probability density. These filters are often referred to as the sigma point Kalman filters or the derivative-free Kalman (local) filters and show improved performance over the EKF. However, there is an important implementation issue that arises in the UKF, particularly in high-dimensional systems. Specifically, the "plain" UKF[10] results in some negative weights for state dimensions greater than 3, which could potentially lead to numerical problems. A notable advantage of the CKF over the UKF is its numerical stability. In particular, according to the numerical stability factor metric defined in Ref.[9], the CKF is more stable with desirable numerical properties. Furthermore, the CKF has better accuracy as seen through its lower root mean square position error and bias norm compared with the UKF.

Some of the recent nonlinear filtering algorithms built using sequential Monte-Carlo-based methods such as particle filters (PF), do provide good performance, but they can be computationally quite expensive.[11,12]

Recently, Pei et al.[13] proposed an efficient nonlinear filtering algorithm called the Gaussian-sum cubature Kalman filter (GSCKF) showing comparable performance to the PF for the bearings-only tracking problem.[13] But the split-merge scheme in GSCKF is not adaptive and the numerical stability of CKF has to be improved. Especially, the case that there is a significant amount of process noise is not considered in GSCKF and it will increase the variance of each term in the Gaussian sum and thereby create a large overlap of the individual Gaussian term.[14]

In this paper we propose the adaptive Gaussian sum based squared-root cubature Kalman Filter with split-merge scheme (AGSSCKF) for the State estimation problem in nonlinear and non-Gaussian systems and compare its performance against GSCKF and other conventional algorithms. The main features of AGSSCKF are the following.

(1) It is built using the squared-root extension of newly introduced cubature Kalman filter (SCKF),[9] which uses the third-degree spherical-radial rule to numerically approximate the multidimensional integral involved in Bayesian filtering.
(2) The initial conditions of the algorithm are similar to that of the Gaussian sum unscented Kalman filter (GSUKF),[15] whereby the initial condition PDF and process noise PDF are given by a Gaussian sum or are approximated by a Gaussian sum using the SMILE method[16] for circumventing the difficulty associated with large plant noise. And each Gaussian component is assigned an independent SCKF with adaptively updated weight.
(3) It consists of an adaptive split-merge procedure, which is designed to deal with highly nonlinear scenarios. Specifically, when a measure of nonlinearity for a Gaussian component exceeds a threshold, it is split into several components adaptively. Thereafter, some components are merged so that the computational costs can be reasonable.

The proposed AGSSCKF demonstrates superior performance, comparable to the PF in both RMSE (root mean square error) and filter consistency, while requiring only a fraction of the computation time needed for the PF.

The paper is organized as follows. Section 2 deals with the state estimation problem and presents a brief description of the SCKF. Then, in Section 3, new scheme for Gaussian components adaptation is designed and discussed and the novel Gaussian sum-based state estimation method is proposed. This section also highlights important practical implementation issues of weight adaption of Gaussian components. The details of the simulations and the comparisons of the performances of the proposed algorithm against several conventional algorithms are given in Section 4. The main contributions of this paper concluding remarks are summarized in Section 5.

2. Squared-root cubature Kalman filter

Consider the following discrete-time nonlinear stochastic system

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \quad (k = 0, 1, 2\ldots) \tag{1}$$

$$z_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \quad (k = 0, 1, 2\ldots) \tag{2}$$

where the vectors $\boldsymbol{x}_k \in \mathbf{R}^{n_x}$ and $z_k \in \mathbf{R}^{n_z}$ represent the unmeasurable state of the system and measurement at time instant k, respectively. n_x and n_z denote the dimensions of state and measurement, respectively. $\boldsymbol{f}(\mathbf{R}^{n_x}) \to \mathbf{R}^{n_x}$ and $\boldsymbol{h}(\mathbf{R}^{n_z}) \to \mathbf{R}^{n_x}$ are known vector mappings, and $\boldsymbol{w}_k \in \mathbf{R}^{n_x}$, $\boldsymbol{v}_k \in \mathbf{R}^{n_z}$ are the state and measurement white noises respectively and are mutually independent.

The PDFs of the noises, $p(\boldsymbol{w}_k)$ and $p(\boldsymbol{v}_k)$ respectively are supposed to be known. The PDF of the initial state $\boldsymbol{x}_0$, $p(\boldsymbol{x}_k)$ is supposed to be known as well, and the initial state is independent from both noises.

The aim of the state estimation is to find the state estimate in the form of the conditional PDF $p(\boldsymbol{x}_k|\boldsymbol{z}^k)$ in which $\boldsymbol{z}^k \triangleq [z_0, z_1, \ldots, z_k]$. In some cases, it suffices to find the first two conditional moments, i.e. the mean $\hat{\boldsymbol{x}}_{k/k} = E[\boldsymbol{x}_k|\boldsymbol{z}^k]$ and covariance matrix $\boldsymbol{P}_{k/k} = \mathrm{cov}[\boldsymbol{x}_k|\boldsymbol{z}^k]$, which can be understood as a Gaussian approximation of conditional PDF, i.e. $p[\boldsymbol{x}_k|\boldsymbol{z}^k] \approx \mathcal{N}\{\boldsymbol{x}_k : \hat{\boldsymbol{x}}_{k/k}, \boldsymbol{P}_{k/k}\}$.[17]

Before specifying a general SCKF algorithm, third-degree spherical-radial rule will be introduced to facilitate transparency of the SCKF algorithm, which is the important theoretical basis of the proposed algorithm in this paper. Under additive Gaussian noise assumption, the state prediction and measurement prediction often require the integration of a nonlinear function with respect to a normal density[9], i.e.,

$$\hat{x}_{k+1/k} = E[x_{k+1}|z^k] = \int_{\mathbf{R}^{n_x}} f(x_k) p[x_{k+1}|z^k] \mathrm{d}x_k \approx \int_{\mathbf{R}^{n_x}} f(x_k) \mathcal{N}\{x_k : \hat{x}_{k/k}, P_{k/k}\} \mathrm{d}x_k \quad (3)$$

$$\hat{z}_{k+1} = E[z_{k+1}|x_{k+1}] \approx \int_{\mathbf{R}^{n_x}} h(x_{k+1}) \mathcal{N}\{x_{k+1} : \hat{x}_{k+1/k}, P_{k+1/k}\} \mathrm{d}x_{k+1} \quad (4)$$

It is well-known that for an arbitrary function $f(x)$, the integral

$$I(f) = \sqrt{2\pi}|\Sigma|^{-1/2} \times \int_{\mathbf{R}^{n_x}} f(x) \exp\left[-\frac{1}{2}(x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)\right] \mathrm{d}x \quad (5)$$

can be expressed in the spherical coordinate system as

$$I(f) = (2\pi)^{-n_x/2} \times \int_{r=0}^{\infty} \left[\int_{U_{n_x}} f(Crz + \mu) \mathrm{d}z\right] r^{n_x-1} \mathrm{e}^{-r^2/2} \mathrm{d}r \quad (6)$$

where $x = Crz + \mu$ with $\|z\| = 1$, μ and Σ is the mean and covariance of x, respectively; C is the Cholesky decomposition of Σ and U_{n_x} is the unit sphere.[9] This integral is further approximated by the symmetric spherical cubature rule.

$$I(f) = \frac{1}{2n_x} \sum_{i=1}^{2n_x} f(\sqrt{n_x}(C\xi_i + \mu)) \quad (7)$$

where n_x is the dimension of x and ξ_i is the ith cubature point located at the intersection of the unit sphere and its axes.[9] The cubature points can be obtained off-line using a third degree centrally symmetric cubature rule.

Then, the resulting squared-root extension of cubature Kalman filter for state estimation problem contains the following steps.[9]

(1) Initialization

Initialization the filter with initiate state $x_{0/0}$, covariance matrix $P_{0/0}$, where $P_{0/0} = S_{0/0} S_{0/0}^{\mathrm{T}}$.

Generate cubature points ξ_i, $i = 1, 2, \ldots, 2n_x$ for dimensional state x_0.

Generate weights $w_i^{\mathrm{c}} = w^{\mathrm{c}} = 1/2n_x$.

(2) Time update ($k = 1, 2, \ldots$)

Evaluate the cubature points

$$X_{i,k-1/k-1} = S_{k-1/k-1}\xi_i + \hat{x}_{k-1/k-1} \quad (8)$$

Evaluate the propagated cubature points

$$X_{i,k/k-1}^{*} = f_k(X_{i,k-1/k-1}, u_{k-1}) \quad (9)$$

Evaluate the predicted state based on the weights

$$\hat{x}_{k/k-1} = \sum_{i=1}^{2n_x} w_i^{\mathrm{c}} X_{i,k/k-1}^{*} \quad (10)$$

Evaluate the squared-root factor of the predicted error

$$S_{k/k-1} = \mathrm{Tria}\left(\left[\chi_{k/k-1}^{*}, S_{Q_{k-1}}\right]\right) \quad (11)$$

where $S_{Q_{k-1}}$ denotes a square-root factor of Q_{k-1} such that $Q_{k-1} = S_{Q_{k-1}} S_{Q_{k-1}}^{\mathrm{T}}$. And the weighted, centered (prior mean is subtracted off) matrix

$$\chi_{k|k-1}^{*} = \frac{1}{\sqrt{2n_x}} \times \left[X_{1,k/k-1}^{*} - \hat{x}_{k/k-1}, \quad X_{2,k/k-1}^{*} - \hat{x}_{k/k-1}, \ldots, X_{2n_x,k/k-1}^{*} - \hat{x}_{k/k-1}\right] \quad (12)$$

It should be noted that $S = \mathrm{Tria}(A)$ denotes a general triangularization (e.g., the QR decomposition) algorithm, where S is a lower triangular matrix. The matrices S and A are related as follows: Let C be an upper triangular matrix obtained from the QR decomposition on A^{T}. Then, we can get an upper triangular matrix $S = C^{\mathrm{T}}$.

(3) Measurement update ($k = 1, 2, \ldots$)

Evaluate the cubature points

$$X_{i,k/k-1} = S_{k/k-1}\xi_i + \hat{x}_{k/k-1} \quad (13)$$

Evaluate the propagated cubature points

$$Z_{i,k/k-1} = h(X_{i,k/k-1}, u_k) \quad (14)$$

Estimate the predicted measurement

$$\hat{z}_{k/k-1} = \frac{1}{2n_x} \sum_{i=1}^{2n_x} Z_{i,k/k-1} \quad (15)$$

Estimate the square-root of the innovation covariance matrix

$$S_{zz,k/k-1} = \mathrm{Tria}([Z_{k/k-1}, S_{R_k}]) \quad (16)$$

where S_{R_k} denotes the square-roots of R_k which is the covariance matrix of measurement noise, and the weighted, centered matrix

$$Z_{k/k-1} = \frac{1}{\sqrt{2n_x}} \times [Z_{1,k/k-1} - \hat{z}_{k/k-1}, Z_{2,k/k-1} - \hat{z}_{k/k-1}, \ldots, Z_{2n_x,k/k-1} - \hat{z}_{k/k-1}] \quad (17)$$

Estimate the cross-covariance matrix

$$P_{xz,k/k-1} = \chi_{k/k-1} Z_{k/k-1}^{\mathrm{T}} \quad (18)$$

where the weighted, centered matrix

$$\chi_{k/k-1} = \frac{1}{\sqrt{2n_x}} \times [X_{1,k/k-1} - \hat{x}_{k/k-1}, X_{2,k/k-1} - \hat{x}_{k/k-1}, \ldots, X_{2n_x,k/k-1} - \hat{x}_{k/k-1}] \quad (19)$$

Estimate the filter gain of SCKF

$$W_k = \left(P_{xz,k/k-1}/S_{zz,k/k-1}^{\mathrm{T}}\right)/S_{zz,k/k-1} \quad (20)$$

Estimate the updated state based on the new measurement z_k

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + W_k(z_k - \hat{z}_{k/k-1}) \quad (21)$$

Estimate the square-root factor of the corresponding error covariance

$$S_{k/k} = \mathrm{Tria}\left(\left[\chi_{k/k-1} - W_k Z_{k/k-1}, W_k S_{R_k}\right]\right) \quad (22)$$

3. Adaptive Gaussian sum filter based on SCKF

SCKF is a new and powerful algorithmic addition to the kit of tools for nonlinear filtering in Gaussian system, especially for high-dimensional nonlinear filtering problem.[9,18] But an initial Gaussian component will become non-Gaussian under nonlinear transformation, increasing the uncertainty propagation error.

To circumvent the state estimation problem in nonlinear and non-Gaussian dynamic system, this paper presents a new Gaussian sum filter (AGSSCKF) based on SCKF with adaptive split-merge scheme. In AGSSCKF, the decrease of the uncertainty propagation error is achieved by splitting the Gaussian component with the largest degree of propagation nonlinearity into multiple ones. At the measurement update stage, when a high degree of nonlinearity is observed for the Gaussian density of a SCKF, it splits into several Gaussian components adaptively. During the filtering step, while the adaptation of the weights can provide us with more accurate uncertainty propagation, this can be further enhanced by judiciously refining or coarsening the Gaussian mixture during propagation. In AGSSCKF, the process of refining Gaussian mixture is triggered by the degree of nonlinearity (DON) observed in each sub-filter because of the nonlinearity of measurement function. At the end of the measurement update stage, some Gaussian components with low weights are pruned or merged for the computation constraints. To deal with the case that there is a significant amount of process noise at the prediction stage, the PDFs of the initial condition and process noise are denoted as the form of Gaussian sum, while SCKF is the sub-filter for each Gaussian component.

Denote $N_{\max}$ as the max number of Gaussian components used as sub-filters in AGSSCKF and N_k as the number at the time instant k. Now, we can proceed and describe our proposed method algorithmically in Table 1 ($\psi_{\text{threshold}}$ denotes the threshold for triggering component splitting).

The steps of the adaptive process of AGSSCKF are described in detail in the following subsections.

Step 1. Initialization

Initiate $N_{\max}$ and $\psi_{\text{threshold}}$, set the time instant $k = 0$ and define a priori initial condition $p(\boldsymbol{x}_0|\boldsymbol{z}^{-1}) = p(\boldsymbol{x}_0)$ as a sum of $N_{k/k-1}$ Gaussian components.

Without loss of generality, suppose the initial condition PDF $p(\boldsymbol{x}_0)$ and state noise PDF $p(\boldsymbol{w}_k)$ are given by a Gaussian sum or are approximated by a Gaussian sum (using split-merge incremental learning method, SMILE[13,16]) with N_0 and q_k terms respectively.

$$p(\boldsymbol{x}_0) = \sum_{m=1}^{N_0} \alpha_0^m \mathcal{N}\{\boldsymbol{x}_0; \hat{\boldsymbol{x}}_0^m, \boldsymbol{P}_0^m\} \tag{23}$$

$$p(\boldsymbol{w}_k) = \sum_{m=1}^{q_k} \beta_k^m \mathcal{N}\{\boldsymbol{w}_k; \hat{\boldsymbol{w}}_k^m, \boldsymbol{Q}_k^m\} \tag{24}$$

where α_0^m, β_k^m are positive weights of particular Gaussian terms with their sum being equal to 1.

$$\begin{cases} \alpha_0^m > 0, & \sum_{m=1}^{N_0} \alpha_0^m = 1 \\ \beta_0^m > 0, & \sum_{m=1}^{q_k} \beta_0^m = 1 \end{cases} \tag{25}$$

All the parameters (i.e. the weights, means and covariance matrices) in initialization step are supposed to be known using SMILE model before performing AGSSCKF.

Step 2. Gaussian component identification

The problem that how to select the critical Gaussian component should be split is discussed in this step.

At the measurement update step of the mth SCKF, we compute the following cubature points to capture the prediction density

$$\boldsymbol{X}_{i,k/k-1}^m = \boldsymbol{S}_{k/k-1}^m \xi_i + \hat{\boldsymbol{x}}_{k/k-1}^m \quad (i = 1, 2, \ldots, 2n_x) \tag{26}$$

where $\hat{\boldsymbol{x}}_{k/k-1}^m$ and $\boldsymbol{S}_{k/k-1}^m$ are the predicted state and covariance, respectively, of the mth filter. Using the spherical-radial rule, the set of cubature points is given as $\xi_i = \sqrt{n_x}\left[\boldsymbol{I}_{n_x} \vdots - \boldsymbol{I}_{n_x}\right]_i$, where $[\boldsymbol{B}]_i$ denotes the ith column of matrix $\boldsymbol{B}$, $\boldsymbol{I}_{n_x}$ is the identity matrix of size n_x and $\vdots$ denotes matrix concatenation. These cubature points are then propagated through the measurement function:

$$\widetilde{\boldsymbol{X}}_{i,k/k-1}^m = h(\boldsymbol{X}_{i,k/k-1}^m) \tag{27}$$

Considering the measurement function $h(\bullet)$ is nonlinear generally, the DON observed in each filter is estimated as the deviation of the propagated cubature points from a linear fit[13,19]:

$$\psi^m \triangleq \frac{1}{n_x} \sum_{t=1}^{n_x} \psi_t^m \tag{28}$$

where

$$\psi_t^m = \frac{1}{2} \| \widetilde{\boldsymbol{X}}_{t,k/k-1}^m + \widetilde{\boldsymbol{X}}_{t+n_x,k/k-1}^m - 2h(\hat{\boldsymbol{x}}_{k/k-1}^m) \|^2 \tag{29}$$

and $1 \leqslant t \leqslant n_x$ because the total number of points required in a SCKF is $2n_x$.

The DON computed by Eq. (28) is used to select the component with the largest possible error reduction after splitting.

Thus the mth Gaussian component to be split is selected using Eq. (30).

$$\psi^m > \psi_{\text{threshold}} \tag{30}$$

Table 1 Algorithm flow of AGSSCKF.

AGSSCKF (performed every time step)
Step 1. Initialization
For $k = 1,2, \ldots$
Step 2. Filtering
while $N_k < N_{\max}$
Estimate DON of each SCKF and find the largest one.
If DON of the mth filter $> \psi_{\text{threshold}}$, then
Step 3. Split the Gaussian density of the mth filter.
else
end while;
end if
end while
Step 4. Measurement update for each SCKF.
Step 5. Global point estimate.
Step 6. Pruning and merging.
Step 7. Prediction.
End for

Step 3. Splitting

Before splitting, the following split-decision model may be used to decide how much sub-components should be split from the most critical Gaussian component m which is given in Step 2.

$$\text{If } \psi_{\text{threshold}} r^{n-1} \leqslant \psi^m < \psi_{\text{threshold}} r^n, \text{ then}$$
$$N_F = 2^n \tag{31}$$

where N_F denotes the number of sub-components after splitting, r is the common ratio of the geometrical progression and $r \geqslant 1$, n is a natural number and $n \geqslant 1$. Obviously, N_F is decided by the relation between r and $\psi_{\text{threshold}}$, which can be selected from empirical training. On the other hand, if $\psi^m < \psi_{\text{threshold}}$, the Gaussian density of the mth filter will not be split, then $N_F = 1$.

The split-decision model given by Eq. (31) means that a critical Gaussian component can be split into 2, 4, 8 or more components according to the degree of nonlinear propagation. This is done by successively splitting the given Gaussian density into 2 components with necessary times in the direction of the eigenvector corresponding to the largest eigenvalue of the covariance matrix. Now, the following splitting scheme may be used to approximate a Gaussian density $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a mixture of two symmetrical Gaussian functions $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$.

$$\begin{cases} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \approx \omega \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\omega)\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \\ \boldsymbol{\mu}_1 = \boldsymbol{\mu} + l\sqrt{\lambda}\boldsymbol{e}_{\text{v}} \\ \boldsymbol{\mu}_1 = \boldsymbol{\mu} - l\sqrt{\lambda}\boldsymbol{e}_{\text{v}} \\ \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma} - l^2\lambda\boldsymbol{e}_{\text{v}}(\boldsymbol{e}_{\text{v}})^{\text{T}} \\ \omega > 0 \end{cases} \tag{32}$$

where λ is the largest eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{e}_{\text{v}}$ is the corresponding eigenvector, l is a displacement parameter that determines the distance between the means of the new Gaussian components and $l = 0.5$ is a good choice suggested in Ref.[20]. The weight ω is set to be 0.5 in most algorithms like GSCKF in Ref.[13] and the split-merge scheme in Ref.[21].

At the end of the splitting procedure, the number of Gaussian components increases by N_F-1.

Step 4. Distributed filtering

After the splitting and optimization stage, the measurement update stage will be performed on each of the Gaussian component for all the sub-filters. Suppose that there are N_f filters for the distributed filtering. Now, new squared-root cubature points are generated to approximate the fth Gaussian components in the prediction density of the mth filter, where $f = 1, 2, \ldots, N_F$.

$$\boldsymbol{X}_{i,k/k-1}^{m,f} = \boldsymbol{S}_{k/k-1}^{m,f}\boldsymbol{\xi}_i + \hat{\boldsymbol{x}}_{k/k-1}^{m,f} \quad (i = 1, 2, \ldots, 2n_x) \tag{33}$$

Then, the particular estimated state $\hat{\boldsymbol{x}}_{k|k}^{m,f}$ and the square-root factor of the corresponding error covariance $\boldsymbol{S}_{k|k}^{m,f}$ of the fth Gaussian component of the mth filter are computed by the SCKF relations presented as Eqs. (14)–(22).

Since the weight of the mth component in the Gaussian mixture before splitting is

$$\alpha_k^m = \frac{p(z_k|\boldsymbol{x}_k, m)\alpha_{k-1}^m}{\sum_{i=1}^{N_k} p(z_k|\boldsymbol{x}_k, i)\alpha_{k-1}^i} \tag{34}$$

where $p(\boldsymbol{z_k}|\boldsymbol{x_k}, m)$ is the likelihood of measurement for the mth filter. Then the updated weight of the fth Gaussian component of the mth filter can be computed by

$$\alpha_k^{m,f} = \frac{p(z_k|\boldsymbol{x}_k, m, f)\alpha_{k-1}^m\alpha_{k-1}^{m,f}}{\sum_{i=1}^{N_k}\sum_{j=1}^{N_f} p(z_k|\boldsymbol{x}_k, i, j)\alpha_{k-1}^i\alpha_{k-1}^j} \tag{35}$$

where $p(z_k|\boldsymbol{x}_k, m, f)$ is the likelihood of measurement for the fth Gaussian component of the mth filter.

Step 5. Global point estimate

After distributed filtering, the global point estimated state and covariance matrix are, respectively, computed as

$$\hat{\boldsymbol{x}}_{k/k} = \sum_{m=1}^{N_k}\sum_{f=1}^{N_f} \alpha_k^{m,f}\hat{\boldsymbol{x}}_{k/k}^{m,f} \tag{36}$$

$$\begin{cases} \boldsymbol{P}_{k/k} = \sum_{m=1}^{N_k}\sum_{f=1}^{N_f}\left\{\alpha_k^{m,f}\left[\boldsymbol{P}_{k/k}^{m,f} + \boldsymbol{b}\boldsymbol{b}^{\text{T}}\right]\right\} \\ \boldsymbol{b} = \boldsymbol{x}_{k/k}^{m,f} - \hat{\boldsymbol{x}}_{k/k}^{m,f} \end{cases} \tag{37}$$

where $\boldsymbol{P}_{k/k}^{m,f} = \boldsymbol{S}_{k/k}^{m,f}(\boldsymbol{S}_{k/k}^{m,f})^{\text{T}}$, then the square-root factor of the corresponding error covariance can be obtained from

$$\boldsymbol{P}_{k/k} = \boldsymbol{S}_{k/k}(\boldsymbol{S}_{k/k})^{\text{T}} \tag{38}$$

Step 6. Merging

The iterative splitting process in the above steps yields a reduction in the uncertainty propagation error by judiciously increasing the order of the Gaussian components. Due to computation constraints, the important procedure continues by judiciously pruning and merging for the reduction of the overall number of Gaussian components.

There are three cases that some components should be merged for the sake of computation spending.

(1) All Gaussian components that maintain low weights before and after a time step integration will be merged gradually.
(2) Two Gaussian components that closely approximate a Gaussian function both before and after a time step integration will be merged.
(3) Two Gaussian components will be merged if their degrees of nonlinear propagation are under a tolerable threshold (i.e. 0.2 $\psi_{\text{threshold}}$) both before and after a time step integration

Suppose that the jth and j'th components should be merged, the merging for the mean, covariance and weight of the newly introduced component is given by equation as follows:

$$\begin{cases} \hat{\boldsymbol{x}}_{k/k}^{jj'} = \sum \alpha_k^q \hat{\boldsymbol{x}}_{k/k}^q \\ \boldsymbol{P}_{k/k}^{jj'} = \sum \alpha_k^q \left[\boldsymbol{P}_{k/k}^q + \left(\hat{\boldsymbol{x}}_{k/k}^q - \hat{\boldsymbol{x}}_{k/k}^{jj'}\right)\left(\hat{\boldsymbol{x}}_{k/k}^q - \hat{\boldsymbol{x}}_{k/k}^{jj'}\right)^{\text{T}}\right] \\ \alpha_k^{jj'} = \alpha_k^j + \alpha_k^{j'} \end{cases} \tag{39}$$

where $q = \{j, j'\}$.

Step 7. Prediction

To deal with the case that there is a significant amount of process noise, the predictive PDF is approximated by the following equation.

$$\begin{cases} p(\boldsymbol{x}_{k+1}|\boldsymbol{z}^k) \approx \sum_{j=1}^{N_{k+1/k}} \alpha^{uj}_{k+1/k} \mathcal{N}\{\boldsymbol{x}_{k+1}; \hat{\boldsymbol{x}}^j_{k+1/k}, \boldsymbol{P}^j_{k+1/k}\} \\ N_{k+1/k} = N_{k/k} q_k \\ \alpha^j_{k+1/k} = \alpha^j_{k/k} \beta^j_k \end{cases} \tag{40}$$

where $N_{k/k}$ is the total number of Gaussian components after pruning and merging, q_k and β^j_k are defined in Eq.(24). Given the filtering estimate mean $\hat{\boldsymbol{x}}^{j'}_{k/k}$, covariance matrix $\boldsymbol{P}^{j'}_{k/k}$, the process noise mean $\hat{\boldsymbol{w}}^l_k$ and covariance matrix $\boldsymbol{Q}^l_k$, the predictive mean $\hat{\boldsymbol{x}}^j_{k+1|k}$ and squared-root of covariance matrix $\boldsymbol{S}^j_{k+1/k}$ are computed by the SCKF relations presented as Eqs.(8)–(12). And the indices j' and l are given by equation as follows:

$$\begin{cases} j' = j - \left\lfloor \dfrac{j-1}{N_{k/k}} \right\rfloor N_{k/k} \\ l = 1 + \left\lfloor \dfrac{j-1}{N_{k/k}} \right\rfloor \end{cases} \tag{41}$$

where $j = 1, 2, \ldots, N_{k+1/k}$, $l = 1, 2, \ldots, q_k$. The symbol $\lfloor a \rfloor$ denotes the floor function, i.e. the largest integer less than or equal to a. The variable j' and l denote the changed indices because of the inconstant Gaussian components in each filtering step.

Until now, the filtering and prediction of AGSSCKF at time instant k are done. Let $k = k + 1$ and the algorithm continues by Step 2.

4. Numerical illustration

In this section, we report the experimental results obtained by applying the AGSSCKF when applied to two nonlinear state estimation problems: nonlinear non-Gaussian system[15] with one-dimensional state (Example 1) and bearings-only tracking problem with two-dimensional state (Example 2).[13]

Performance of the following state estimation methods was compared in the two numerical examples: Standard global filters, such as GSUKF, GSCKF, PF, AGSSCKF. Local filters, such as SCKF, UKF.

The PF used is the regularized sequential importance resampling filter and 1000 particles are used, while the scaling parameter κ in the UKF and GSUKF is selected such that $\kappa + n_x = 3$, as suggested in Ref.[7]. $N_{\max} = 15$ independent SCKFs are used in the AGSSCKF and The independent SCKFs are given equal and normalized weights during filter initialization. The threshold of nonlinearity $\psi_{\text{threshold}}$ in Eq. (30) is set to be 0.02 and the common ratio in Eq. (31) is defined as $r = 1.5$.

Several standard performance metrics are used in the performance analysis of our numerical illustration to compare the accuracy and consistency of the different state estimation algorithms discussed. The different performance metrics used are described as follows:

(1) RMSE

$$\text{RMSE}_k = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[\hat{\boldsymbol{x}}_{k/k}(i) - \boldsymbol{x}_k(i)\right]^2} \tag{42}$$

where $\boldsymbol{x}_k(i)$, $\hat{\boldsymbol{x}}_{k|k}(i)$ are the true position and estimated position respectively at time k of the ith Monte Carlo run, and M is the total number of Monte Carlo runs.

(2) Normalized estimation error squared(NESS)

NEES is a measure to check for filter consistency and is defined as follows.[22,23]

$$\varsigma = (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k/k})^{\mathrm{T}} \boldsymbol{P}^{-1}_{k/k} (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k/k}) \tag{43}$$

where $\boldsymbol{x}_k$, $\hat{\boldsymbol{x}}_{k|k}$ are the true position and estimated position respectively and $\boldsymbol{P}_{k|k}$ is the updated state covariance at time step k.

The consistency of a filter is the ability of the filter to accurately estimate uncertainty. An "inconsistent" filter produces estimation errors that are larger than those predicted by the model on which the estimator is based.

(3) Computational complexity

The complexities of the state estimate algorithms investigated are compared in terms of the relative computational running time required for the simulations.

4.1. One-dimensional state estimation

(1) Simulation scenario and filter initialization

Consider the nonlinear non-Gaussian system with one-dimensional state

$$\boldsymbol{x}_{k+1} = \phi_1 \boldsymbol{x}_k + \sin(\omega\pi k) + \boldsymbol{w}_k \tag{44}$$

where the process noise $\boldsymbol{w}_k$ is described by Gamma PDF $Ga(3, 2)$ and $\forall k$, $\phi_1 = 0.5$, $\omega = 0.04$. The state is observed by the scalar measurement described by the equation

$$\boldsymbol{z}_k = \begin{cases} \phi_2 \boldsymbol{x}_k^2 + \boldsymbol{v}_k & (k \leqslant 30) \\ \phi_3 \boldsymbol{x}_k - 2 + \boldsymbol{v}_k & (k > 30) \end{cases} \tag{45}$$

where the measurement noise $\boldsymbol{v}_k$ is described by Gaussian PDF $\mathcal{N}\{\boldsymbol{v}_k{:}0, 10^{-5}\}$ and $\forall k$, $\phi_2 = 0.2$, $\phi_3 = 0.5$. The initial state condition is given by a sum of five Gaussian PDFs

$$p(\boldsymbol{x}_0) = \sum_{j=1}^{5} 0.2 \times \mathcal{N}\{\boldsymbol{x}_0 : j-3, 10\} \tag{46}$$

and the predictive PDF $p(\boldsymbol{x}_0|\boldsymbol{z}^{-1}) = p(\boldsymbol{x}_0)$.

On the other hand, $\forall k$, a three-term Gaussian sum approximation of the $Ga(3, 2)$ distribution by means of the SMILE model is calculated as

$$\begin{aligned} \tilde{p}(\boldsymbol{w}_k) = {} & 0.29\mathcal{N}\{\boldsymbol{w}_k : 2.14, 0.72\} + 0.18\mathcal{N}\{\boldsymbol{w}_k \\ & : 7.45, 8.05\} + 0.53\mathcal{N}\{\boldsymbol{w}_k : 4.31, 2.29\} \end{aligned} \tag{47}$$

which means $\beta^1_k = 0.29, \beta^2_k = 0.18, \beta^3_k = 0.53$ in the Eq. (25).

(2) Performance analysis

Time progresses of the RMSE and average NEES for a few selected filtering methods in one MC run are illustrated in Fig. 1. After a number of 100 MC runs, the average RMSE (ARMSE), average NEES (ANEES) and computational costs for all $k \leqslant 30$ are given in Table 2 and the dimension of computational costs in a time step is megasecond (ms).

When $k \leqslant 30$, the simulation results demonstrate a substantial increase of the estimate quality (comparable to PF) of Gaussian sum filter with an adaptive split-merge scheme to the standard global filtering methods like GSUKF and

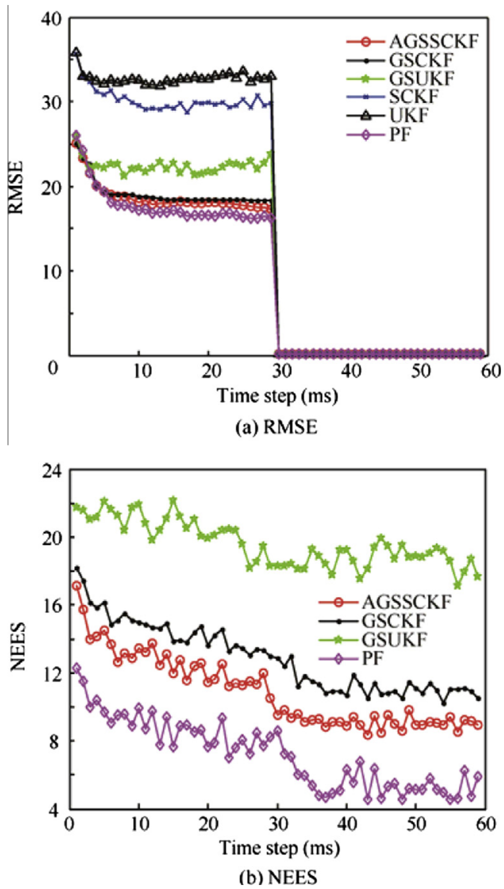

Fig. 1 Time development of the RMSE and NEES of several methods in Example 1.

Table 2 Performance comparison in Example 1.

Method	ARMSE	ANEES	Time (ms)
UKF	33.25	433.34	0.85
SCKF	30.12	390.52	0.78
GSUKF	23.22	20.22	8.96
GSCKF	18.86	15.38	8.93
AGSSCKF	18.05	13.56	11.62
PF	16.32	10.67	96.64

GSCKF. In terms of NEES, AGSSCKF and PF show a balanced behavior as opposed to GSUKF and GSCKF which are overly optimistic. However, the NEES curves of UKF and CKF are not analyzed in the Fig. 1 because they line too much higher than that of other methods.

When $30 < k \leqslant 60$, the RMSE has a rapid decrease from the time instant 30 on Fig. 1, which is due to the change in the measurement Eq. (45) which becomes linear.

As far as the computational costs are concerned, utilizing the splitting procedure in AGSSCKF leads to an increase of the costs. However, if an effective simplified procedure (pruning and merging) is used, the increase is approximately 20% only compared to GSUKF and GSCKF. On the other side, the split-decision model has a negligible effect on computational costs.

4.2. Bearings-only tracking

Next we review a classical filtering application in which we track a moving object with sensors, which measure only the bearings (or angles) of the object with respect to positions of the sensors. There is a one moving target in the scene and two angular sensors for tracking it. Solving this problem is important, because often more general multiple target tracking problems can be partitioned into sub-problems, in which single targets are tracked separately at a time.

(1) Simulation scenario and filter initialization

The state vector in two-dimensional target tracking system can be expressed as

$$\boldsymbol{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^{\mathrm{T}} \tag{48}$$

where x_k, y_k denote the position of target in two-dimensional cartesian coordinates and the velocity toward those coordinate axes is denoted as $\dot{x}_k, \dot{y}_k$.

The dynamics of the target is modeled as a linear, discretized Wiener velocity model:

$$\boldsymbol{x}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \boldsymbol{q}_{k-1} \tag{49}$$

where Δt denotes sampling interval and $\boldsymbol{q}_k$ is the process noise. $\forall k$, a three-term Gaussian sum approximation of the distribution of $\boldsymbol{q}_k$ by means of the SMILE model is calculated as

$$\begin{aligned} \tilde{p}(\boldsymbol{q}_k) = {} & 0.55\mathcal{N}\{\boldsymbol{q}_k; 1.23, 7.56\} + 0.33\mathcal{N}\{\boldsymbol{q}_k; 0.26, 5.32\} \\ & + 0.12\mathcal{N}\{\boldsymbol{q}_k; 0.18, 2.02\} \end{aligned} \tag{50}$$

The measurement model for sensor i is defined as

$$\theta_k^i = \arctan\frac{y_k - s_y^i}{x_k - s_x^i} + r_k^i \tag{51}$$

where (s_x^i, s_y^i) is the position of sensor i and $(s_x^1, s_y^1) = (-1, -2), (s_x^2, s_y^2) = (1, 1)$, the measurement noise is described by Gaussian PDF as $r_k^i \sim \mathcal{N}(0, \delta^2)$, with $\delta = 0.1$ radians.

The target starts with state $\boldsymbol{x}_0 = [0,0,1,0]^{\mathrm{T}}$, and in the estimation we set the prior distribution for the state to $\boldsymbol{x}_0 \sim \mathcal{N}(0, \boldsymbol{P}_0)$, where

$$\boldsymbol{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \tag{52}$$

Fig. 2 shows the bearings-only tracking scenario in this simulation.

(2) Performance analysis

The RMSE and NEES results of the different algorithms for bearing only tracking scenarios in one MC run are shown in Fig. 3. After a number of 100 MC runs, the ARMSE, ANEES and Computational costs of a time step (500 time step in all) are given in Table 3.

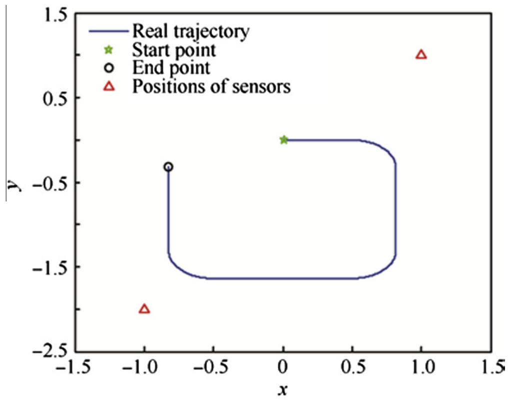

Fig. 2 Bearings only tracking scenario.

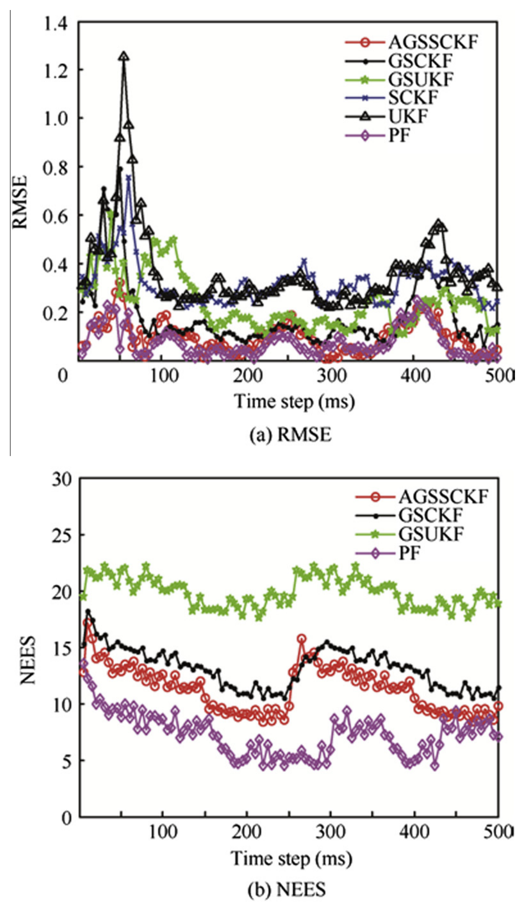

Fig. 3 Time development of RMSE and NEES of several methods in Example 2.

From the above simulation results, the RMSE of the CKF is similar to that of the UKF. The RMSE curves of the GSCKF and GSUKF overlap the AGSSCKF RMSE curve, which nearly matches the curve of PF. What's more, the AGSSCKF successfully brings the performance gap closer to PF, especially when the target has a curved trajectory.

For the same reason, the NEES curves of UKF and CKF do not appear in Fig. 3, which means that these two algorithms have poor consistency. On the other side, PF shows excellent consistency in this simulation. The three Gaussian sum-based filters have improved NEES performance compared with their original version. With the adaptive split-merge scheme, the NEES of AGSSCKF is slightly lower than other two algorithms, which shows an effective enhancement in the aspect of filter consistency.

Table 3 Performance comparison in Example 2.

Method	ARMSE	ANEES	Time (ms)
UKF	0.1532	246.47	0.92
SCKF	0.1530	215.35	0.83
GSUKF	0.1203	18.33	9.92
GSCKF	0.1056	8.91	11.23
AGSSCKF	0.9234	6.96	13.22
PF	0.8021	4.82	120.64

In terms of computational cost, the PF runs much slower than other existing algorithms as it requires a large number of particles in its implementation to get good filtering performance. CKF runs faster than UKF because it requires only $2n_x$ cubature points compared with $2n_x + 1$ sigma points in UKF. The three Gaussian sum-based algorithms introduce different extra computational times, but they still show a major computational advantage over PF.

However, the optimal selection of threshold value $\psi_{\text{threshold}}$ to perform splitting is scenario dependent and will be investigated in future work.

5. Conclusions

(1) An efficient new algorithm called AGSSCKF is developed in this paper for state estimation in nonlinear non-Gaussian discrete dynamic systems. The simulation results exhibit that the estimate quality of AGSSCKF is comparable to the particle filter with significantly reduced computational cost.
(2) The initial conditions and the process noise are denoted in the form of Gaussian sum in AGSSCKF and a bank of Squared-root cubature Kalman filter is used as the sub-filters to estimate state of system with the corresponding adaptively updated weights respectively.
(3) AGSSCKF consists of an adaptive splitting and merging scheme to handle difficult highly nonlinear cases and a split-decision model is proposed based on the nonlinearity degree of measurement for the improvement of splitting performance.
(4) The same criteria and adaptation procedures can be used for any Gaussian sum-based global estimation method dependent on local filters.

Our future work in this topic will include development of a systematic procedure to determine the optimum scheme for splitting and merging and the weights optimization of Gaussian components after splitting and merging procedure.

Acknowledgments

The authors thank the associate editor and the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China

(No. 61032001) and Shandong Provincial Natural Science Foundation of China (No. ZR2012FQ004).

References

1. Yin JJ, Zhang JQ, Zhuang ZS. Gaussian sum PHD filtering algorithm for nonlinear non-Gaussian models. *Chin J Aeronaut* 2008;**21**(1):341–51.
2. Dimitri PB. *Dynamic programming and optimal control*. 3rd ed. Nashua: Athena Scientific; 2007. p. 323–33.
3. Wang X, Han CZ. An improved multiple model GM-PHD filter for maneuvering target tracking. *Chin J Aeronaut* 2013;**26**(1): 179–85.
4. Brian DA, John BM. *Optimal filtering*. New Jersey: Dover Publications; 2005. p. 121–34.
5. Nardone SC, Lindgren AG, Gong KF. Fundamental properties and performance of conventional bearings-only target motion analysis. *IEEE Trans Autom Control* 1984;**29**(9):775–87.
6. Magnus N, Niels KP, Ole R. New developments in state estimation for nonlinear systems. *Automatica* 2000;**36**(11): 1627–38.
7. Simon JJ, Jeffrey KU. Unscented filtering and nonlinear estimation. *Proc IEEE* 2004;**92**(3):401–21.
8. Ito K, Xiong K. Gaussian filters for nonlinear filtering problems. *IEEE Trans Autom Control* 2000;**45**(5):910–27.
9. Arasaratnam I, Haykin S. Cubature kalman filters. *IEEE Trans Autom Control* 2009;**54**(6):1254–69.
10. Wan E, Vander MR. *The unscented Kalman filter*. Hoboken: Wiley Publishing; 2001. p. 221–80.
11. Vander MR, Wan E. Sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. *Proceedings of the international conference on acoustics, speech, and signal processing*; 2003 April 6–10; Hong Kong, China; 2003. p. 701–4.
12. Šimandl M. Straka O. Nonlinear filtering methods: some aspects and performance evaluation. *International conference on modelling, identification and control*; 2003 Feb 10–13; Innsbruck, Austria; 2003. p. 67–72.
13. Pei HL, Sanjeev A, Tharaka AL, Thushara DA. A Gaussian-sum based cubature Kalman filter for bearings-only tracking. *IEEE Trans Aerosp Electron Syst* 2013;**49**(2):1161–76.
14. Daniel LA, Harold WS. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans Autom Control* 1972;**17**(4):439–48.
15. Straka O, Duník J, Šimandl M. Gaussian sum unscented Kalman filter with adaptive scaling parameters. *Proceedings of international conference on information fusion*; 2011 July 5–8; Chicago, UK; 2011. p. 1413–20.
16. Konstantinos B, Isaac EL. Spilt-merge incremental learning (SMILE) of mixture models. *International conference on artificial neural networks*; 2007 Sep 9–13; Porto, Portugal; 2007. p. 291–300.
17. Sorenson HW, Alspach DL. Recursive Bayesian estimation using Gaussian sums. *Automatica* 1971;**7**(4):465–79.
18. Sun F, Tang LJ. Estimation precision comparison of cubature Kalman filter and unscented Kalman filter. *Control Decis* 2013;**28**(2):303–8 [Chinese].
19. Faubel F, Klakow D. An adaptive level of detail approach to nonlinear estimation. *Proceedings of the international conference on acoustics, speech, and signal processing*; 2010 Mar 14–19; Dallas, UK; 2010. p. 3958–61.
20. Faubel F, Klakow D. Further improvement of the adaptive level of detail transform: Splitting in direction of the nonlinearity. *Proceedings of the 18th European signal processing conference*; 2010 Aug 23–27; Aalborg, Denmark; 2010. p. 850–4.
21. Terejanu G. An adaptive split-merge scheme for uncertainty propagation using Gaussian mixture models. *The 49th AIAA aerospace sciences meeting*; 2011 Jan 4–7; Orlando, UK; 2011. p. 1–8.
22. Bailey T, Nieto J, Nebot E. Consistency of the fast SLAM algorithm. *Proceedings of the IEEE international conference on robotics and automation*; 2006 May 15–19; Orlando, UK; 2006. p. 424–9.
23. Vander H. Consistency checks for particle filters. *IEEE Trans Pattern Anal Mach Intell* 2006;**28**(1):140–5.

Liu Yu received his M.S. and Ph.D. degrees from Naval Aeronautical and Astronautical University in 2010 and 2014, respectively. He is now a researcher in the Institute of Information Fusion, Naval Aeronautical and Astronautical University. His main research interests are information fusion and nonlinear filtering.

Dong Kai is a Ph.D. candidate at Naval Aeronautical and Astronautical University, where he received his M.S. degree in 2009. His area of research includes information fusion and multi-target tracking.