Neurocomputing 158 (2015) 155–166

Contents lists available at ScienceDirect

Neurocomputing

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

# Observation noise modeling based particle filter: An efficient algorithm for target tracking in glint noise environment ☆

Hangyuan Du [a,*], Wenjian Wang [a,b], Liang Bai [a,b]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, PR China
[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, Shanxi 030006, PR China

ARTICLE INFO

*Article history:*
Received 8 May 2014
Received in revised form
9 January 2015
Accepted 27 January 2015
Communicated by M. Bianchini
Available online 7 February 2015

*Keywords:*
Target tracking
Particle filter
Observation likelihood
Glint noise
Gaussian mixture model

ABSTRACT

In this paper, a novel particle filtering algorithm for target tracking in the presence of glint noise based on observation noise modeling is proposed. The algorithm samples particles using the observation likelihood function, the construction of which is converted to a modeling problem of observation noise. Additionally, the Gaussian mixture model is incorporated to approximate the distribution of observation noise at each time instant. In order to derive a recursive form update for the parameters of the Gaussian components, the maximum likelihood estimation method is employed, enabling noise to be effectively tracked by fusing the latest observations. The algorithm is then used in simulations of bearings-only tracking problems in a glint noise environment with two types of targets: non-maneuvering and maneuvering. The results of the proposed algorithm are evaluated and compared to several existing filtering algorithms through a series of Monte Carlo simulations. The simulation results demonstrate that the proposed algorithm is more precise, robust, and even has a faster convergence rate than the comparative filters. Lastly, the performance of the proposed filter in situations with different numbers of particles and Gaussian components is explored using the simulation results.



## 1. Introduction

Target tracking is the process of maintaining a state estimation for a moving target based on a set of sensor measurements [1]. In the decades since its establishment as a unique scientific field, the study of target tracking has continued to evolve with the development of sensor system. The Bayes filter provides a recursive state estimation framework for target tracking by using probability theory. In practice, there are a lot of realizations for Bayes filter of which the Kalman filter (KF) is the most common one. The KF is capable of achieving optimal solutions to estimation problems of the linear systems with Gaussian noise [2], though unfortunately, very few actual target tracking problems fall into this category in practice. Recently, the particle filter (PF), a non-parametric random sampling algorithm that can address more complicated target tracking problems, has begun to attract an increasing amount of attention with the improvement of computing capacity, since recursive Bayes filter can be designed for non-linear non-Gaussian state estimation problem through the use of the Monte Carlo method [3]. However, in the glint noise disturbed tracking system, the particle filter does not work very well. The present work proposes an improvement for this issue.

Unlike the KF and its variants (e.g., the Extended Kalman filter (EKF), the Unscented Kalman filter (UKF), the Divided Difference Filter (DDF) and the Gauss–Hermite filter (GHF) [4,5]), the PF is a Sequential Monte Carlo (SMC) algorithm. The PF is also referred to as the sequential importance sample (SIR) algorithm that possesses a strong processing capacity for non-linear and non-Gaussian problems. It uses a group of randomly sampled particles to describe the posterior probability distribution of state. The posterior estimation is obtained by propagating the weighted particles through a dynamic state space (DSS) in the recursive process, and estimation results are reliable only when a sufficiently large number of particles are sampled [3]. In theory, the computational complexity of the PF is dependent only on the number of particles used and the design of the algorithm itself, and not influenced by the dimension of the state. Additionally, the structure of the PF is highly suited to parallel computing.

Since its conception, a number of comprehensive studies have been conducted on how to improve PF accuracy and expand its application possibilities. A commonly employed strategy is to improve the design of the importance density function from which particles are sampled [6]. In some studies, a suboptimal estimator such as the EKF, UKF or central difference filter (CDF) is used to

☆This work is supported by the National Natural Science Foundation of China (Nos. 61305073, 61273291, 41401521), and Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (No. 2014104).

* Corresponding author.
*E-mail address:* duhangyuan@sxu.edu.cn (H. Du).

http://dx.doi.org/10.1016/j.neucom.2015.01.057
0925-2312/

fuse the latest observation and generate importance density, which significantly improves the estimation accuracy of the filter [7–9]. In the literature [5], accuracy of the filter is improved by sampling particles from an annealing form of a priori transition density. MacCormick [10] proposed a partitioned sampling method for optimizing the design of the importance density function when the likelihood reveals the feature of sharp shape and has little overlap with the prior transition density. In another study [11], the authors provide an optimization-based algorithm (using the steepest descent method) that was used to generate the importance density and then sample particles from the density.

Another problem constraining the performance of the PF is the particle degeneracy phenomenon, where after several iterations, all but one particle will have negligible weight. However, this issue can be avoided by using resampling, prompting many researchers to investigate in resampling strategies. Aside from the most common resampling methods, such as multinomial resampling, systematic resampling, residual resampling and stratified resampling [12], there are many innovative PFs based on the exploitation of the resampling operation. The auxiliary particle filter (APF) introduces an auxiliary variable that identifies the component index of each particle, and exchanges sampling and resampling sequences [13]. The adaptive PF incorporates the Kullback–Leibler distance (KLD) method into either sampling or resampling to ensure that the measure of error between the estimation and real posterior densities does not exceed a pre-specified error bound. Within this algorithm, the number of particles changes, adapting according to the signal environment [14,15]. Other studies focus on particle impoverishment which occurs when very few particles have significant weights while most other particles with small weights are discarded in resampling. For example, Markov Chain Monte Carlo (MCMC) is used to diversify particles [16,17], which weakens particle correlation and makes the distribution of particle set more stationary. The regularized particle filter (RPF) extracts samples from the continuous approximate probability distribution, and alleviates the impoverishment problem to a degree by using a kernel function [16,18]. In [19], an exquisite resampling (ER) algorithm was introduced into the PF, which enabled the impoverishment phenomenon to be avoided and for the estimation accuracy to be obtained using a smaller number of particles.

Additional tactics for enhancing the PF include the Rao-Blackwellized PF (RB-PF) [20,21], the Gaussian Sum PF (GSPF) [22,8,23], the intelligent optimization based PF [24,25], and the distributed PF [26,27]. All of these algorithms improve the estimation accuracy of PF and expand its scope of application through different means.

None of the aforementioned PFs, however, can adequately execute state estimation in a non-linear system with the assumption of non-Gaussian non-stationary observation noise, such as glint noise in radar tracking. Therefore, in this paper, a novel particle filter is developed to address this issue. The filter selects the observation likelihood function as the importance density function, and transforms the construction of likelihood function into an observation noise modeling problem. Then, the probability density function (PDF) of the observation noise is modeled as a bank of weighted Gaussian densities where the distribution parameters of individual Gaussian components are calculated in each iteration using the maximum likelihood estimation (MLE) in each iteration. This noise model and the calculated distribution parameters are subsequently used to build the observation likelihood function for extracting particles. In simulations, the proposed algorithm performs very well in radar target tracking problems with glint noise.

The rest of paper is organized as follows. Section 2 gives a brief review of the preliminary. In Section 3, the novel likelihood PF based on the Gaussian mixture noise model is proposed. The construction of observation likelihood function is transformed into an estimation problem of the PDF of the observation noise, which is approximated as a bank of Gaussian components. Calculation formulas for the distribution parameters of these noise components are also deduced. Section 4 describes the implementation of the proposed algorithm as well as computational complexity analysis. In Section 5, the proposed algorithm is applied using two object tracking simulations. Finally, conclusions are drawn in Section 6.

## 2. Preliminary

### 2.1. Glint noise with non-Gaussian non-stationary distribution

Glint noise, or angle glint, which is a major disturbance in radar tracking systems refers to fluctuations in the measured angles of arrival of backscattered electromagnetic waves transmitted from a radar device. It arises due to interference between two or more reflections from the target surface which induces a distortion in the shape of the propagating wave front and angular errors [28]. These angular errors corrupt the sensors line-of-sight measurements, especially in bearings-only object tracking systems. Fig. 1 presents a typical record of disturbed angular observations collected by a BQM-34A [29] from which we can easily see that the distribution of glint noise is decidedly different from that of Gaussian noise. First, there are a number of random spikes distributed throughout the record, indicating that the sophisticated uncertainty distribution would be heavy tailed. Second, the statistical process is non-stationary, that is, the probability distribution changes over time throughout the entire observation history. In order to obtain an accurate and effective state estimation for tracking, a credible statistical model for glint noise must be established. However, due to its non-stationary statistic properties, building an accurate prior probability model for glint noise using a direct parametric description is difficult. In some studies, glint noise has been approximated as one type of common distribution or as combination of types. For example, Ref. [30] modeled the glint noise as a Student's $t$ distribution with two degrees of freedom; Ref. [31] used prior Gamma distribution to estimate parameters of the Student's $t$ distribution model for the glint noise; Refs. [28] and [29] approximated glint noise as a combination of two Gaussian distributions; and in [32], glint noise was described as a combination of Gaussian and Laplacian distributions.

The aforementioned noise models have one thing in common: they can describe the non-Gaussian and heavy tailed properties of glint noise very well, however, they all belong to a class of noise models that use a priori knowledge. That is to say, the parameters of the noise model must be set (via experience or experiment) before the filtering process, and they lack online adjustability. As a


**Fig. 1.** A typical angular observation record disturbed by glint noise.

result, these models ignore the non-stationary property of glint noise, making them incapable of accurately describing the noise statistic as it changes over time. To achieve a better tracking performance with non-Gaussian non-stationary noise, the dynamics of the noise statistic must be monitored in real-time during the filtering process.

### 2.2. Target tracking in the context of Bayes filtering

Target tracking problems are typically described using the DSS method [33] in which the motion model and the observation model are given by the following equations:

$$\begin{cases} \mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{u}_k, \\ \mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k, \end{cases} \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^{d_x}$ is an unobserved system state vector at time instant $k$, $\mathbf{z}_k \in \mathbb{R}^{d_z}$ is an observation obtained by the sensor, and $\mathbf{u}_k \in \mathbb{R}^{d_u}$ and $\mathbf{v}_k \in \mathbb{R}^{d_v}$ are random noises with given or unknown probability distributions, accompanying the system update and sensor observation, respectively. $f_k(\cdot)$ and $h_k(\cdot)$ are, respectively, the state transition function and the observation function. The task of target tracking is to estimate the state of $\mathbf{x}_k$ at each time instant through the continued acquisition and processing of observation $\mathbf{z}_k$ with the given initial state $\mathbf{x}_0$. Usually, the history sequence of states and observations is denoted as $\mathbf{x}_{0:k} \equiv \{\mathbf{x}_0, \ldots, \mathbf{x}_k\}$ and $\mathbf{z}_{0:k} \equiv \{\mathbf{z}_0, \ldots, \mathbf{z}_k\}$, respectively. Within a Bayes filter framework, the estimation of the tracking task is divided into two stages—– time update and state update.

In time update stage, the prior density of the state at time instant $k$ is calculated as [5]

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1}, \tag{2}$$

where $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ is referred to as the prior transition density. The prior density is then updated to the posterior density during the state update stage by fusing latest observations accordingly:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = C_k p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}), \tag{3}$$

where $p(\mathbf{z}_k \mid \mathbf{x}_k)$ is the likelihood probability and the normalization coefficient $C_k$ can be described as

$$C_k = \left( \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})\, \mathrm{d}\mathbf{x}_k \right)^{-1}. \tag{4}$$

### 2.3. Sampling importance resampling particle filter

The PF propagates a set of particles that have been randomly sampled from the posterior density in space. But in most cases, sampling directly from the posterior distribution is very difficult or impossible because the data are usually sparse in high-dimensional space and the region of interest is relatively small in the whole data space. Therefore, the posterior is typically replaced by an importance density function, known as the sequential importance sampling (SIS) algorithm, which has a known distribution that is easy to sample from [16]. On this basis, the resampling procedure is carried out, the particle degeneracy problem is overcome, and the effectiveness of particle set is ensured by removing particles with smaller weights and duplicating particles with larger weights. Once resampling is introduced, the algorithm as a whole is referred to as the sampling importance resampling particle filter, or SIR PF [6]. The pseudocode of SIR PF can be represented as follows:

*Step* 1: Sample $N$ particles from the importance density function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k)$:

$$\mathbf{x}_k^j \sim q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k), \quad j = 1, \ldots, N. \tag{5}$$

*Step* 2: Calculate the importance weight of each particle:

$$\omega_k^j \propto \omega_{k-1}^j \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^j) p(\mathbf{x}_k^j \mid \mathbf{x}_{k-1}^j)}{q(\mathbf{x}_k^j \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k)}, \tag{6}$$

where $\omega_k^j$ is the weight of particle $\mathbf{x}_k^j$, $p(\mathbf{z}_k \mid \mathbf{x}_k^j)$ and $p(\mathbf{x}_k^j \mid \mathbf{x}_{k-1}^j)$ denote observation likelihood and state prior transition functions, respectively.

*Step* 3: Normalize all the weights:

$$\omega_k^j = \frac{\omega_k^j}{\sum_{j=1}^N \omega_k^j}. \tag{7}$$

*Step* 4: Calculate effective sample size:

$$\hat{N}_{eff} = \frac{1}{\sum_{j=1}^N \left(\omega_k^j\right)^2}. \tag{8}$$

Compare $\hat{N}_{eff}$ with the predefined threshold value $N_{th}$. If $\hat{N}_{eff} < N_{th}$, this indicates that there are some invalid particles in the set, and that resampling should be performed.

*Step* 5: Export the sum of the weighted particles, and estimate the posterior probability density of the system state as follows:

$$p(\mathbf{x}_k \mid \mathbf{z}_k^j) \approx \sum_{j=1}^N \omega_k^j \cdot \delta(\mathbf{x}_k - \mathbf{x}_k^j), \tag{9}$$

where the Kronecker Delta function $\delta(\bullet)$ is described as

$$\delta(t-\tau) = \begin{cases} 0, & t \neq \tau, \\ 1, & t = \tau \end{cases}. \tag{10}$$

### 2.4. Likelihood particle filter

The estimation performance of the PF is significantly dependent on the selection of the importance density function, so the optimal importance density function should have the least possible weight variances. Theorem 1, given below, provides a possible optimal importance density function [34]:

**Theorem 1.** *$q_{opt}(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k)$ is an optimal importance density function based on minimized importance weight variances $\mathrm{Var}_{q(\mathbf{x}_k \mid \mathbf{x}_k^j, \mathbf{z}_k)}(\omega_k^j)$ of $\mathbf{x}_{k-1}^j$ and $\mathbf{z}_k$, and the importance weight is updated as*

$$\omega_k^j = \omega_{k-1}^j p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^j). \tag{11}$$

However, the optimal importance density function presented in Theorem 1 has two problems. Firstly, at times, it may be result in a non-standard distribution sometimes that is difficult to sample from. And second, in order to update importance weight according to (11), the integral operation described in (12) which is always non-analytic is necessary:

$$p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^j) = \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j)\, dx. \tag{12}$$

In practice, a widely used importance density function is the prior state transition density:

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^j, \mathbf{z}_k) = p(\mathbf{x}_k^j \mid \mathbf{x}_{k-1}^j). \tag{13}$$

At this point, the sampling process is easy to implement, and the importance weight is updated as follows:

$$\omega_k^j = \omega_{k-1}^j p(\mathbf{z}_k \mid \mathbf{x}_k^j). \tag{14}$$

However, the prior density does not fuse the latest observation information, so, the sampled particles may drift off the real posterior distribution, resulting in a degradation of performance. Several experimental results have shown that the observation

likelihood usually makes a greater contribution to posterior distribution and is more similar to posterior density than it is to prior density [3]. Thus, by extracting particles from the observation likelihood rather than the prior state transition density, in a method known as likelihood particle filter, the PF can be expected to concentrate on particles in the peak region of the likelihood function, which will improve estimation performance [35].

Unfortunately, there is still a problem that the observation likelihood may be non-analytic in many cases. Considering this problem, Ref. [36] introduced a piecewise approximation method, in which the observation likelihood was divided into several uniform intervals according to sampling number, and the center of each interval was selected as representative point. Then, the likelihood value of each interval was approximated using the value of the representative point, and the whole observation likelihood is obtained by combining these piecewise approximations. The method works well with a single-dimensional likelihood, but in multi-dimensional problems, interval partitioning may be difficult. In this paper, we exploit the relationship between the observation likelihood and the probability distribution of observation noise and seek to construct a common model for observation likelihood, the details of which are presented in Section 3.

## 3. Constructing observation likelihood for the particle filter

In this paper, the observation likelihood function, from which the particles are extracted in the likelihood PF, depends on the observation noise. In this way, we are able to generate the observation likelihood by modeling non-Gaussian non-stationary observation noise recursively within the particle filter framework.

### 3.1. The relationship between observation likelihood and observation noise

The observation likelihood function can be approximated as Gaussian distribution density as (15), under the assumption that observation noise represented by $\mathbf{v}_k$, is zero mean Gaussian white noise, with a variance of $R_k$:

$$p(\mathbf{z}_k|\mathbf{x}_k^j) = N(\mathbf{z}_k; h_k(\mathbf{x}_k^j), R_k). \tag{15}$$

Unfortunately, with regard to glint noise with non-Gaussian non-stationary properties, the observation likelihood cannot be approximated using Gaussian distribution without introducing excess error into the estimation process. To overcome this issue, we will design an alternative approximation method by exploring the important relationship between the observation noise and the observation likelihood.

Assume that $a$, $b$, and $c$ are three independent random variables that can be described as follows:

$$a+b=c. \tag{16}$$

Based on the conclusion in [37], we have

$$p_{c|a}(c|a) = p_b(c-a). \tag{17}$$

If we consider (1), we get

$$p(\mathbf{z}_k|\mathbf{x}_k) = p(\mathbf{z}_k - h_k(\mathbf{x}_k)) = p(\mathbf{v}_k). \tag{18}$$

For each particle, we define the noise sample as follows:

$$\mathbf{v}_k^j = \mathbf{z}_k - h(\mathbf{x}_k^j), \tag{19}$$

and the observation likelihood of the $j$-th particle can be described as

$$p(\mathbf{z}_k|\mathbf{x}_k^j) = p(\mathbf{v}_k^j). \tag{20}$$

By exploiting this result, the construction of the observation likelihood is transformed into an observation noise modeling problem.

### 3.2. Glint noise model based on a Gaussian mixture

If $\mathbf{v}_k$ is considered to be non-Gaussian non-stationary glint noise with a non-zero mean, the PDF of $\mathbf{v}_k$ at the instant $k$ in time can be described as a Gaussian mixture composed of a bank of weighted Gaussian components with size $K$:

$$p(\mathbf{v}_k) = \sum_{i=1}^{K} \alpha_{i,k} p_i(\mathbf{v}_k) = \sum_{i=1}^{K} \alpha_{i,k} N(\mathbf{v}_k; \mu_{i,k}, \delta_{i,k}^2), \tag{21}$$

where $p_i(\mathbf{v}_k)$ denotes the PDF of the $i$-th Gaussian component at instant $k$, and $\alpha_{i,k} > 0$, $\mu_{i,k}$ and $\delta_{i,k}$ are the weight, mean and standard deviation, respectively, of the individual components. Also,

$$\sum_{i=1}^{K} \alpha_{i,k} = 1. \tag{22}$$

For each noise sample, we have

$$p\{\mathbf{v}_k^j | p_i(\mathbf{v}_k^j)\} = p_i\{\mathbf{v}_k^j | \mu_{i,k}, \sigma_{i,k}^2\}. \tag{23}$$

Let the event $O_{i,k}$ denote that the $j$-th noise sample $\mathbf{v}_k^j$ is identically distributed with the $i$-th Gaussian component:

$$O_{i,k} = [\mathbf{v}_k^j \sim N\{\mu_{i,k}, \sigma_{i,k}^2\}]. \tag{24}$$

Additionally, the probability of the event $O_{i,k}$ is

$$P\{O_{i,k}\} = \alpha_{i,k}. \tag{25}$$

The probability of the $i$-th Gaussian component conditioned on the $j$-th noise sample can be calculated by using the Bayes rule such that

$$P\left\{p_i(\mathbf{v}_k^j)|\mathbf{v}_k^j\right\} = \frac{P\{p_i(\mathbf{v}_k^j), \mathbf{v}_k^j\}}{P\left\{\left(\mathbf{v}_k^j\right)\right\}} = \frac{p\{\mathbf{v}_k^j | p_i(\mathbf{v}_k^j)\} P\{p_i(\mathbf{v}_k^j)\}}{\sum_{i=1}^{K} p\left\{\mathbf{v}_k^j | p_i(\mathbf{v}_k^j)\right\} P\left\{p_i\left(\mathbf{v}_k^j\right)\right\}}, \tag{26}$$

where $p\{\mathbf{v}_k^j | p_i(\mathbf{v}_k^j)\}$ is PDF of $\mathbf{v}_k^j$ with a given $p_i(\mathbf{v}_k^j)$, and because of (25), we have

$$P\{p_i(\mathbf{v}_k^j)\} = P\{O_{i,k}\} = \alpha_{i,k}. \tag{27}$$

Therefore substituting (23) and (27) into (26), we obtain

$$P\left\{p_i(\mathbf{v}_k^j)|\mathbf{v}_k^j\right\} = \frac{\alpha_{i,k} p_i\left\{\mathbf{v}_k^j | \mu_{i,k}, \sigma_{i,k}^2\right\}}{\sum_{i=1}^{K} \alpha_{i,k} p_i\left\{\mathbf{v}_k^j | \mu_{i,k}, \sigma_{i,k}^2\right\}}. \tag{28}$$

### 3.3. Calculation of distribution parameters for Gaussian components

The key issue of observation noise modeling is acquiring the distribution parameters of each Gaussian component, $\mu_{i,k}$, $\sigma_{i,k}^2$, and $\alpha_{i,k}$. In this paper, these parameters are divided into two classes at each time instant: prior distribution parameters $(\alpha_{i,k^-}, \mu_{i,k^-}, \sigma_{i,k^-}^2)$ and posterior distribution parameters $(\alpha_{i,k^+}, \mu_{i,k^+}, \sigma_{i,k^+}^2)$, and they are alternately calculated in the iterative filtering process. First, the prior parameters are obtained using the time update from the posterior parameters at the previous time instant. Then, they are transformed to posterior parameters through the state update by fusing with the acquired observations. Through this process, a posteriori PDF approximation of the observation noise is obtained.

#### 3.3.1. Selection of prior distribution parameters

The prior distribution parameters of each component in the mixture model at time instant $k$ are selected as according to the following:

For each component, prior weight is initialized as

$$\alpha_{i,k^-} = \frac{1}{K}. \tag{29}$$

When $K$ is odd, the prior mean is set as

$$\mu_{i,k^-} = \overline{\mu}_{k-1}\left(1+\left(i-\frac{K-1}{2}\right)\frac{1}{K}\right), \tag{30}$$

and if $K$ is even, the prior mean is set as

$$\mu_{i,k^-} = \overline{\mu}_{k-1}\left(1+\left(i-\frac{K}{2}\right)\frac{1}{K}\right), \tag{31}$$

where

$$\overline{\mu}_{k-1} = \sum_{i=1}^{K} \alpha_{i,k-1}\mu_{i,k-1}. \tag{32}$$

The prior variance is

$$\sigma^2_{i,k^-} = \sigma^2_{i,k-1}. \tag{33}$$

*3.3.2. Calculation of posteriori distribution parameters*

By fusing the latest observations, we can update the prior distribution parameters to the posteriori stage. This process can be realized by maximizing the likelihood function of all the noise samples conditioned on the Gaussian mixture model. For the convenience of computing, the logarithm of the likelihood function is used as a substitute. The logarithmic likelihood function is given by [38]

$$L_k = \sum_{j=1}^{N} \ln \sum_{i=1}^{K} \alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}. \tag{34}$$

Letting $\lambda$ be a Lagrange multiplier, the Lagrange function can be established as follows by obeying the constraint in (22):

$$L_k = \sum_{j=1}^{N} \ln \sum_{i=1}^{K} \alpha_{i,k} p_i\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\} - \lambda\left(\sum_{i=1}^{K}\alpha_{i,k} - 1\right). \tag{35}$$

The posterior parameters of each Gaussian component can then be calculated by letting the partial derivatives of the Lagrange function be zero:

(1) *Mean*: The mean $\mu_{i,k}$ of the $i$-th Gaussian component can be obtained by letting $\partial L_k/\partial\mu_{i,k} = 0$:

$$\frac{\partial L_k}{\partial \mu_{i,k}} = \sum_{j=1}^{N} \frac{\alpha_{i,k}\frac{\partial}{\partial\mu_{i,k}} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}. \tag{36}$$

Considering (23), Eqs. (36) leads to

$$\begin{aligned}\frac{\partial L_k}{\partial \mu_{i,k}} &= \sum_{j=1}^{N} \frac{\alpha_{i,k}\frac{\partial}{\partial\mu_{i,k}}\left\{\frac{1}{\sqrt{2\pi\sigma^2_{i,k}}}\exp\left[-\frac{\left(\mathbf{v}_k^j-\mu_{i,k}\right)^2}{2\sigma^2_{i,k}}\right]\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}} \\ &= \sum_{j=1}^{N} \frac{\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}\frac{\mathbf{v}_k^j-\mu_{i,k}}{\sigma_{i,k}}.\end{aligned} \tag{37}$$

Using (28) and (37), we acquire

$$\frac{\partial L_k}{\partial \mu_{i,k}} = \sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}\frac{\mathbf{v}_k^j-\mu_{i,k}}{\sigma_{i,k}} = 0. \tag{38}$$

Therefore,

$$\mu_{i,k} = \frac{\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}\mathbf{v}_k^j}{\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}}. \tag{39}$$

(2) *Variance*: The variance $\sigma_{i,k}$ of the $i$-th Gaussian component can be obtained by letting $\partial L_k/\partial\sigma_{i,k} = 0$:

$$\begin{aligned}\frac{\partial L_k}{\partial \sigma_{i,k}} &= \sum_{j=1}^{N} \frac{\alpha_{i,k}\frac{\partial}{\partial\sigma_{i,k}}\left\{\frac{1}{\sqrt{2\pi\sigma^2_{i,k}}}\exp\left[-\frac{\left(\mathbf{v}_k^j-\mu_{i,k}\right)^2}{2\sigma^2_{i,k}}\right]\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}} \\ &= \sum_{j=1}^{N} \frac{\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}\left[\frac{\left(\mathbf{v}_k^j-\mu_{i,k}\right)^2}{\sigma^3_{i,k}} - \frac{1}{\sigma_{i,k}}\right].\end{aligned} \tag{40}$$

Using (28), we obtain

$$\frac{\partial L_k}{\partial \sigma_{i,k}} = \sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}\frac{\left[\left(\mathbf{v}_k^j-\mu_{i,k}\right)^2-\sigma^2_{i,k}\right]}{\sigma^3_{i,k}} = 0. \tag{41}$$

Therefore,

$$\sigma^2_{i,k} = \frac{\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}\left(\mathbf{v}_k^j-\mu_{i,k}\right)^2}{\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}}. \tag{42}$$

(3) *Weight and Lagrange multiplier*: Similarly, weight of each component can be obtained by solving $\partial L_k/\partial\alpha_{i,k} = 0$:

$$\frac{\partial L_k}{\partial \alpha_{i,k}} = \sum_{j=1}^{N} \frac{p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}}{\sum_{i=1}^{K}\alpha_{i,k} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k}, \sigma^2_{i,k}\right\}} - \lambda. \tag{43}$$

Substituting (28) into (43), we acquire

$$\frac{\partial L_k}{\partial \alpha_{i,k}} = \sum_{j=1}^{N} \frac{P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}}{\alpha_{i,k}} - \lambda = 0. \tag{44}$$

Then, solving (44), we obtain

$$\alpha_{i,k} = \frac{1}{\lambda}\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}. \tag{45}$$

Also,

$$\begin{aligned}\sum_{i=1}^{K}\alpha_{i,k} &= \sum_{i=1}^{K}\frac{1}{\lambda}\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\} \\ &= \frac{1}{\lambda}\sum_{i=1}^{K}\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\} \\ &= \frac{K}{\lambda} = 1,\end{aligned} \tag{46}$$

such that

$$\lambda = K. \tag{47}$$

Substituting the above equation into (45), we get

$$\alpha_{i,k} = \frac{1}{K}\sum_{j=1}^{N} P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\}. \tag{48}$$

For each Gaussian component in the noise mixture model, the posterior means, variances and weights can be calculated using Eqs. (39), (42) and (48), respectively, where

$$P\left\{p_i\left(\mathbf{v}_k^j\right) \mid \mathbf{v}_k^j\right\} = \frac{\alpha_{i,k^-} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k^-}, \sigma^2_{i,k^-}\right\}}{\sum_{i=1}^{K}\alpha_{i,k^-} p_i\left\{\mathbf{v}_k^j \mid \mu_{i,k^-}, \sigma^2_{i,k^-}\right\}}.$$

## 4. Improved likelihood particle filter

In this study, we extract particles from the observation likelihood which is transformed into the PDF of observation noise. In

this way, the corresponding importance weight is updated as

$$\begin{aligned}\omega_k^j &= \omega_{k-1}^j \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^j\right)p\left(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j\right)}{q\left(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j,\mathbf{z}_k\right)}\\ &= \omega_{k-1}^j \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^j\right)p\left(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j\right)}{p\left(\mathbf{z}_k|\mathbf{x}_k^j\right)}\\ &= \omega_{k-1}^j p\left(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j\right).\end{aligned} \tag{49}$$

### 4.1. Implementation of the new filter

The detailed algorithm flow of the proposed PF is summarized as follows:

1. Initialization: $k=0$

*Step* 0: For $j=1,\ldots,N$, draw particles $\mathbf{x}_0^j$ from the initial importance density function, for example the prior transition density or any initial observation likelihood functions. Then, initialize the Gaussian components for the observation noise model.

2. For $k=1,2,\ldots$

*Step* 1: For the $K$ Gaussian components in the observation noise model at each time instant, the prior distribution parameters are estimated firstly. The prior weight is

$$\alpha_{i,k^-} = \frac{1}{K},$$

and if $K$ is odd, the prior mean is

$$\mu_{i,k^-} = \overline{\mu}_{k-1}\left(1+\left(i-\frac{K-1}{2}\right)\frac{1}{K}\right).$$

Otherwise, if $K$ is even

$$\mu_{i,k^-} = \overline{\mu}_{k-1}\left(1+\left(i-\frac{K}{2}\right)\frac{1}{K}\right),$$

where

$$\overline{\mu}_{k-1} = \sum_{i=1}^{K}\alpha_{i,k-1}\mu_{i,k-1}.$$

Additionally, the variance is

$$\sigma_{i,k^-}^2 = \sigma_{i,k-1}^2.$$

*Step* 2: Calculate the observation noise samples as follows:

$$\mathbf{v}_k^j = \mathbf{z}_k - h(\mathbf{x}_{k^-}^j),$$

where

$$\mathbf{x}_{k^-}^j = f(\mathbf{x}_{k-1}^j).$$

The conditional probability of each noise Gaussian component can be evaluated by the prior distribution parameters:

$$P\left\{p_i\left(\mathbf{v}_k^j\right)|\mathbf{v}_k^j\right\} = \frac{\alpha_{i,k^-}\,p_i\left\{\mathbf{v}_k^j|\mu_{i,k^-},\sigma_{i,k^-}^2\right\}}{\sum_{i=1}^{K}\alpha_{i,k^-}\,p_i\left\{\mathbf{v}_k^j|\mu_{i,k^-},\sigma_{i,k^-}^2\right\}}.$$

*Step* 3: Update the distribution parameters of the Gaussian components from the prior stage to the posterior stage:

The posterior weight of each Gaussian component is

$$\alpha_{i,k^+} = \frac{1}{K}\sum_{j=1}^{N}P\left\{p_i\left(\mathbf{v}_k^j\right)|\mathbf{v}_k^j\right\}.$$

The posterior mean is

$$\mu_{i,k^+} = \frac{\sum_{j=1}^{N}P\left\{p_i\left(\mathbf{v}_k^j\right)|\mathbf{v}_k^j\right\}\mathbf{v}_k^j}{\sum_{j=1}^{N}P\left\{p_i\left(\mathbf{v}_k^j\right)|\mathbf{v}_k^j\right\}}.$$

And the posterior variance is

$$\sigma_{i,k^+}^2 = \frac{\sum_{j=1}^{N}P\left\{p_i(\mathbf{v}_k^j)|\mathbf{v}_k^j\right\}\left(\mathbf{v}_k^j-\mu_{i,k^+}\right)^2}{\sum_{j=1}^{N}P\left\{p_i\left(\mathbf{v}_k^j\right)|\mathbf{v}_k^j\right\}}.$$

*Step* 4: The observation likelihood is built as

$$p(\mathbf{z}_k|\mathbf{x}_k^j) = p(\mathbf{v}_k^j) = \sum_{i=1}^{K}\frac{\alpha_{i,k^+}}{\sqrt{2\pi\sigma_{i,k^+}^2}}\exp\left[-\frac{\left(\mathbf{v}_k^j-\mu_{i,k^+}\right)^2}{2\sigma_{i,k^+}^2}\right].$$

*Step* 5: Sampling $N$ particles from the observation likelihood:

$$\mathbf{x}_k^j \sim p\left(\mathbf{z}_k|\mathbf{x}_k^j\right),\quad j=1,\ldots,N.$$

*Step* 6: The importance weight of each particle is calculated as

$$\omega_k^j = \omega_{k-1}^j p\left(\mathbf{x}_k^j|\mathbf{x}_{k-1}^j\right).$$

*Step* 7: Normalize the importance weight:

$$\omega_k^j = \frac{\omega_k^j}{\sum_{j=1}^{N}\omega_k^j}.$$

*Step* 8: Perform resampling procedure

*Step* 9: Results outputting

### 4.2. Computational complexity

In this section, the computational complexity of the proposed algorithm is discussed from a theoretical point of view, by clarifying the amount of time required by the algorithm. In any PF, the computational complexities of sampling, weight updating and resampling are proportional to the number of particles. As a result, a generic PF algorithm requires $O(N)$ as a basic computational load [39]. The primary difference between the proposed algorithm and a generic PF is that the construction of the observation likelihood function, i.e. Step 1–Step 4, increases the computational load. In Step 1, the prior distribution parameters of each Gaussian component are initialized, which costs $O(K)$ time. In Step 2, we calculate the noise samples and the conditional probability, operations that needs $O(N+NK^2)$. In Step 3, updating the prior parameters of the Gaussian components to the posterior stage takes $O(NK)$ time. In Step 4, computational load of $O(NK)$ is needed to construct the observation likelihood for sampling. Therefore, the total time complexity required by the proposed algorithm is $O(K+N(1+K+K^2))$. This is linearly related to the number of particles, and has a quadratic relationship to the number of Gaussian components. However, because the amount of Gaussian components in the noise model is much less than the sampled particles, we may conclude that the proposed algorithm improves the filter's performance as the computational load added by it is reasonable. Furthermore, several solutions exist for saving computational cost, including dividing the algorithm into several stages and introducing time-sharing processing or parallel computing.

## 5. Simulation results

In order to examine its tracking capability in the presence of glint noise, the proposed filter has been tested using a series of Monte Carlo simulations of bearings-only target tracking problems [40]. Two scenarios are used for the simulations. First, the performance of the proposed filter is investigated in a non-maneuvering target tracking problem. Then, the simulation is extended to a more complex maneuvering target tracking problem. In addition to the proposed PF, various filtering algorithms,

such as EKF [4], SIR PF [20], likelihood PF [41] and their improved forms [42–46], are used for comparison.

### 5.1. Simulation model

The system model of the moving target is

$$\mathbf{X}_k = \boldsymbol{F}\mathbf{X}_{k-1} + \boldsymbol{G}\mathbf{w}_k, \tag{50}$$

where $\mathbf{X}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^{\mathrm{T}}$ is the state variable, $x_k$ and $y_k$ denote the location of target in the Cartesian system, $\dot{x}_k$ and $\dot{y}_k$ are the velocity components in two directions. $\boldsymbol{F}$ denotes the state transition matrix. Under the assumption of uniform rectilinear motion, it is defined as follows:

$$\boldsymbol{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{51}$$

where $T$ represents the observation cycle of sensor. While in anti-clockwise or clockwise coordinated turn motion, it is defined as

$$\boldsymbol{F} = \begin{bmatrix} 1 & \sin(\varphi_k T)/\varphi_k & 0 & -(1-\cos(\varphi_k T))/\varphi_k \\ 0 & \cos(\varphi_k T) & 0 & -\sin(\varphi_k T) \\ 0 & (1-\cos(\varphi_k T))/\varphi_k & 1 & \sin(\varphi_k T)/\varphi_k \\ 0 & \sin(\varphi_k T) & 0 & \cos(\varphi_k T) \end{bmatrix}, \tag{52}$$

where $\varphi_k$ denotes the turn angular speed ($\varphi_k > 0$ corresponds to anti-clockwise motion, while $\varphi_k < 0$ corresponds to clockwise motion). $\boldsymbol{G}$ is the noise gain matrix, which is given by

$$\boldsymbol{G} = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}. \tag{53}$$

Process noise $w_k$ is zero-mean Gaussian distributed with the covariance $Q$, where $Q = qI_2$, and $I_2$ denotes a $2 \times 2$ identity matrix. With the uniform rectilinear motion, $q=3$ and, with the turn maneuvering motion, $q=5$.

A sensor located at $(0, 0)$ is used to measure the angle between the target and the $X$-axis with an observation cycle of $T=1$ s, The observation model is described as

$$\mathbf{Z}_k = \arctan\frac{y_k}{x_k} + \mathbf{v}_k, \tag{54}$$

where $\mathbf{v}_k$ denotes the glint noise present at each angular observation. By the nature of the experimental design, the observation noise sequence used in the simulation is a purely deterministic process. As a consequence, it is not appropriate to treat the noise sequence as the realization of a stochastic process. Instead, we use the noise sequence as the basis for a simulation of a real glint noise stochastic process. We generate a sampled stochastic process of 150 observations of glint noise from an original data set, which is plotted in Fig. 2. As the figure shows, the mean of this sampled series is 0.0191 rad, and the standard deviation is 0.0489 rad. The maximum and minimum glint errors are 0.2745 rad and −0.1491 rad, respectively.

As mentioned above, there are two different simulation scenarios:

*Non-maneuvering motion*: The target performs a uniform rectilinear motion traveling at a constant velocity (25 m/s, 5 m/s) from a specific position (−50 m, 200 m). The whole simulation continues for 150 time samples, and the initial error covariance matrix is $P_0 = \mathrm{diag}(10, 1, 10, 1)$. Comparisons between the tracking performances of six filters are performed: (1) the EKF; (2) the conventional SIR PF, where particles are sampled from the prior transition probability; (3) the conventional likelihood PF; (4) the differential evolution particle filter (DE-PF) [42], which introduces the differential evolution based resampling scheme; (5) the enhanced likelihood particle filter (EL-PF) [43], where a normalized likelihood function is constructed, which is further exponentially weighted to enhance the weight of the particles in high likelihood areas; and (6) the proposed PF.

*Maneuvering motion*: The target is initially located at the position (100 m, 500 m), and it moves rectilinearly at a constant velocity (20 m/s, −10 m/s) for the first 50 samples. Then for the following 50 samples, the target successively performs an anti-clockwise and then a clockwise turn with the angular speeds of −0.1225 rad/s and −0.1225 rad/s, respectively. Finally, the target moves in uniform rectilinear fashion for 50 samples. In the maneuvering target tracking problem, it is impossible for a single motion to model describe all of the changes of various motion states. For this reason, the interacting multiple model (IMM) is introduced. In the IMM, the shift of motion pattern is expressed in terms of homogeneous Markov chains, and the weighted filtering results of all the single motion models are combined into an end product of the entire process [44]. In this paper, the IMM model set contains one uniform motion model and one turning motion model. The filters used for comparison in simulations consist of (1) the IMM-EKF; (2) the IMM-PF [45]; (3) the IMM-likelihood PF; (4) the Improved IMM-PF for glint noise [46] in which the weight updates for the particles are calculated by a non-linear glint noise PDF which is modeled by a combination of Gaussian and Student's $t$ distributions; (5) the IMM-EL-PF, where the EL-PF is integrated with IMM; and (6) the IMM version of the proposed PF.

For narrative convenience, all of the PF algorithms used in the simulations are listed as Table 1 under the categories of non-maneuvering and maneuvering tracking problems and their simplified representations are given. To guarantee the validity of the following experiment, all the PFs except for the EL-PF employ the

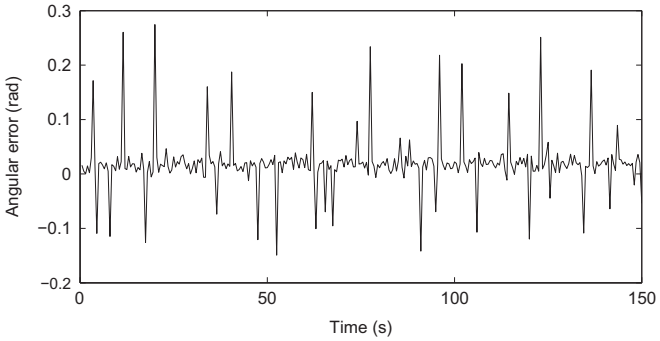


**Fig. 2.** A series of sampled observation glint noises used in simulations.

**Table 1**
Simplified representations of particle filters used in simulations.

| Simulation scenario | Particle filtering | Simplified representation |
|---|---|---|
| Non-maneuvering motion | Conventional SIR PF | PF 1 |
| | Conventional likelihood PF | PF 2 |
| | DE-PF | PF 3 |
| | Enhanced likelihood PF | PF 4 |
| | The proposed PF | PF 5 |
| Maneuvering motion | IMM-PF | PF 1 |
| | IMM-likelihood PF | PF 2 |
| | Improved IMM-PF for glint noise | PF 3 |
| | IMM-EL-PF | PF 4 |
| | IMM version of the proposed PF | PF 5 |

threshold method and the systematic resampling algorithm. The threshold value of the effective sample size is set as $N_{th} = N/2$.

### 5.2. Performance comparison and results analysis

To reduce errors caused by randomness, all the filters are tested in 100 Monte Carlo simulations with same initial conditions. The configurations of the PC used to run the simulation are Inter Core2 E7500 2.93 GHz, 2 GB RAM, and all programs coded by Matlab R2012b. Root mean square error (RMSE) and root time averaged mean square (RTAMS) are used to evaluate estimation accuracy. The RMSE of $\mathbf{x}_k$ is defined as

$$\mathrm{RMSE}_{\mathbf{x}_k} = \sqrt{\frac{1}{L} \times \sum_{i=1}^{L} \left(\mathbf{x}_{k,i} - \hat{\mathbf{x}}_{k,i}\right)^2}, \tag{55}$$

where $L$ denotes the number of Monte Carlo simulations, and $\mathbf{x}_{k,i}$ and $\hat{\mathbf{x}}_{k,i}$ are the true value and the estimation of state variable, respectively. The RTAMS of the state variable throughout the entire simulation sequence for each filter is defined as

$$\mathrm{RTAMS}_{\mathbf{x}} = \sqrt{\frac{1}{M} \times \sum_{k=1}^{M} \mathrm{MSE}_{\mathbf{x}_k}}, \tag{56}$$

where $M$ denotes the time duration of the simulation sequence, and $\mathrm{MSE}_{\mathbf{x}_k}$ is the mean square error ($\mathrm{MSE}_{\mathbf{x}_k} = \mathrm{RMSE}^2_{\mathbf{x}_k}$) of $\mathbf{x}_k$.

#### 5.2.1. Example I: non-maneuvering target tracking

First, the numbers of particles in PF 1 ~PF 5 are set as $N=100$, and the number of components in Gaussian mixture noise model of the proposed filter is set as $K=5$. For each algorithm, the RMSE of the position and velocity estimation of the moving target at each time instant along the $X$ and $Y$ directions are presented in Fig. 3. From the figure, we can see that the performance of EKF suffers strongly in the presence of glint noise, and has the largest estimation error. Compared to EKF, PF 1 and PF 2 improve the estimation accuracy remarkably, however, the errors are still large. Additionally, none of the estimation errors of these three filters are stable, as the statistic of observation noise changes frequently. Further improvements in both estimation accuracy and stability are obtained with PF 3 and PF 4 due to the countermeasures against glint noise developed within the filters. Compared to the other filters, PF 5 has the highest excellent estimation accuracy throughout the simulation sequence. In particular, the proposed filter has a higher convergence rate than any of the other filters. These strong results can be attributed to the fact that the noise model of the proposed filter can describe the true situation of noise probability distribution exactly in real time exactly and that the acquired importance density function is close to the real posterior distribution.

For easier comparison, the estimation errors of all the PFs with different numbers of particles are given in Fig. 4. The figure presents the algorithms' position estimation RTAMS along both the $X$ and $Y$ directions in the form of bar graph. With the same number of particles, the proposed algorithm significantly outperforms other particle filters in terms of estimation RTAMS. As the number of particles in the simulations increases, the errors of all filters trend down, and the proposed filter always keeps the highest estimation accuracy. Lastly, it is worth noting that the estimation errors of all filters change little when the number of particles increasing from 10,000 to 500,000, indicating that the estimation accuracy does not improve endlessly as the number of particles increases, but plateaus once a certain number of particles have reached.

For a more explicit quantitative comparison, the estimation errors and execution times of the different PF algorithms in simulations with 100 particles are listed in Table 2 in which "PF 5($K$)" denotes the proposed algorithm with $K$ Gaussian components in the noise model. From the table, we can see that the proposed filter estimates system states with the highest accuracy and has the longest running time. An additional improvement in accuracy can be obtained by increasing the number of Gaussian components, but unfortunately, this incurs a sharp rise in computational cost, the superabundant Gaussian components are very complicated to initialize, and the improvement is only minor.

The estimation errors of the new filter with different numbers of Gaussian components and particles are presented in Table 3.

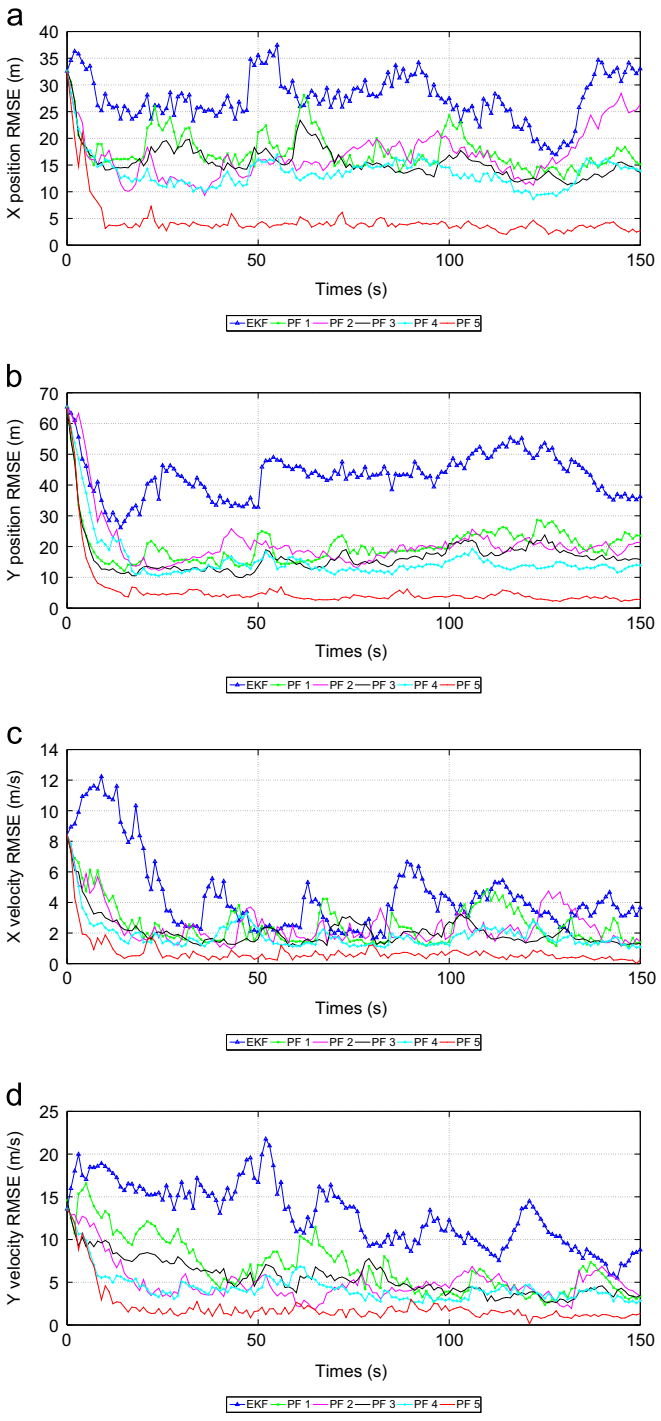


**Fig. 3.** Estimation errors of different filtering algorithms in non-maneuvering target tracking: (a) $X$ position RMSE, (b) $Y$ position RMSE, (c) $X$ velocity RMSE, and (d) $Y$ velocity RMSE.

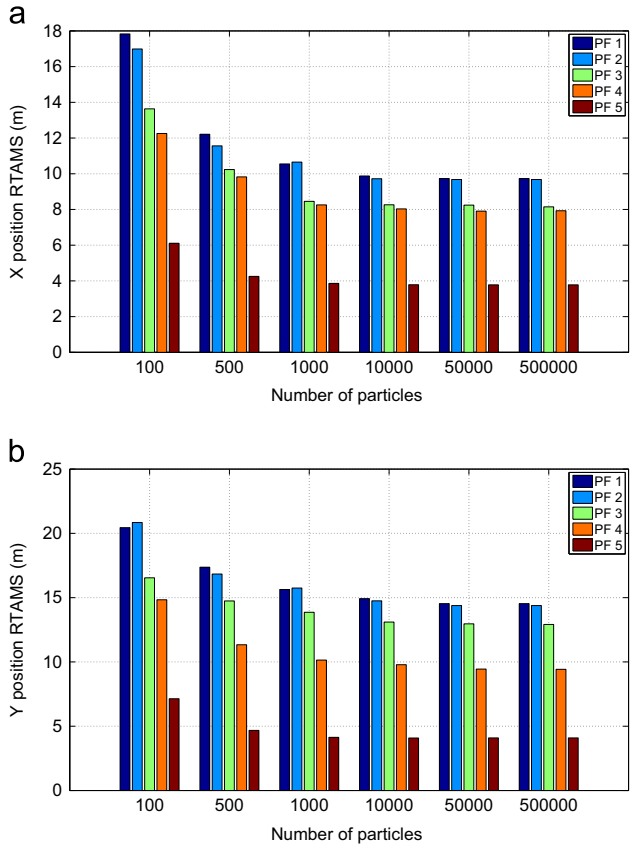


**Fig. 4.** Position estimation RTAMS of different PF algorithms with different particle numbers in non-maneuvering target tracking: (a) *X* position RTAMS and (b) *Y* position RTAMS.

**Table 2**
Performances of different PF algorithms in non-maneuvering tracking simulations with 100 particles.

| Filtering algorithm | *X* position RTAMS (m) | *Y* position RTAMS (m) | *X* velocity RTAMS (m/s) | *Y* velocity RTAMS (m/s) | Execution time (s) |
|---|---|---|---|---|---|
| PF 1 | 17.8309 | 20.4471 | 2.5717 | 6.9235 | 0.2752 |
| PF 2 | 16.9856 | 21.1419 | 2.4636 | 5.0679 | 0.2796 |
| PF 3 | 13.6352 | 16.5426 | 1.9654 | 4.6143 | 0.3716 |
| PF 4 | 12.2546 | 14.8324 | 1.7648 | 3.9179 | 0.6266 |
| PF 5(5) | 6.1080 | 7.1396 | 1.2535 | 2.2783 | 0.7735 |
| PF 5(10) | 5.8263 | 6.8684 | 1.2447 | 2.2435 | 2.2418 |
| PF 5(20) | 5.7741 | 6.8542 | 1.2436 | 2.2452 | 5.2247 |
| PF 5(100) | 5.7826 | 6.8471 | 1.2442 | 2.2433 | 61.8237 |

The table shows that the number of particles has a greater impact on the performance of the new filter within a specific range. When there are fewer particles, the estimation performance can be improved significantly by increasing the sampling number, whereas increasing the number of Gaussian components causes only a marginal improvement, indicating that the developed filter can effectively cope with noise using a small number of Gaussian components.

### 5.2.2. *Example II: maneuvering target tracking*

First, we compare estimation performances of the proposed filter to several other filters under the simulation conditions $N=100$ and $K=5$. The estimation RMSEs of different filters at each time instant are presented in Fig. 5. From the figure, it can be seen that, with their IMM versions, all filters are capable of obtaining convergence results for maneuvering tracking in IMM version. However, IMM-EKF, PF 1, and PF 2 track position and velocity with relatively larger errors and lower convergence rates in the presence of glint noise. PF 3 and PF 4 make limited improvements in overcoming the adverse impacts of glint noise. In contrast, with the same initializations, PF 5 performs very well throughout the simulation sequence, with high accuracy, fast convergence rate, and robustness. These results are consistent with what is indicated in the non-maneuvering target tracking simulation.

**Table 3**
Estimation RTAMS of the new PF algorithm with different numbers of particles and Gaussian components in non-maneuvering target tracking simulations.

| Number of particles | Number of Gaussian components | *X* position RTAMS (m) | *Y* position RTAMS (m) | *X* velocity RTAMS (m/s) | *Y* velocity RTAMS (m/s) |
|---|---|---|---|---|---|
| 100 | 5 | 6.1080 | 7.1396 | 1.2535 | 2.2783 |
| 100 | 10 | 5.8263 | 6.8684 | 1.2447 | 2.2435 |
| 100 | 20 | 5.7741 | 6.8542 | 1.2436 | 2.2452 |
| 100 | 100 | 5.7826 | 6.8471 | 1.2442 | 2.2433 |
| 200 | 5 | 5.4635 | 6.3524 | 0.8726 | 1.4436 |
| 200 | 10 | 5.3364 | 6.3361 | 0.8672 | 1.4388 |
| 200 | 20 | 5.3342 | 6.3142 | 0.8647 | 1.4356 |
| 200 | 100 | 5.3315 | 6.0257 | 0.8641 | 1.4352 |
| 500 | 5 | 4.2524 | 4.6741 | 0.6156 | 0.9125 |
| 500 | 10 | 4.2471 | 4.6752 | 0.6068 | 0.9104 |
| 500 | 20 | 4.2458 | 4.6734 | 0.6024 | 0.9075 |
| 500 | 100 | 4.2462 | 4.6726 | 0.6053 | 0.9071 |
| 1000 | 5 | 3.8596 | 4.1354 | 0.5452 | 0.7806 |
| 1000 | 10 | 3.8485 | 4.0587 | 0.5437 | 0.7784 |
| 1000 | 20 | 3.8466 | 4.0742 | 0.5433 | 0.7765 |
| 1000 | 100 | 3.8474 | 4.0718 | 0.5441 | 0.7762 |

Fig. 6 illustrates the performances of the five PFs in position tracking with different particle numbers. The figure shows that the estimation errors of these filters demonstrate a downward trend as particle number increases within a specific range and that the proposed filter achieves the lowest estimation errors in all simulations.

Accuracy and execution time comparisons for the five PF algorithms with 100 particles are presented in Table 4. The results demonstrate that the proposed filter achieves the optimal estimation accuracy for both position and velocity, even though it has the largest computational load. Table 5 lists the estimation results of the new filter with different numbers of particles and Gaussian components. Similar to the results for non-maneuvering tracking, the accuracy of the filter is primarily dependent on the number of particles within a certain range, whereas increasing the Gaussian components has little effect on estimation accuracy. The estimation errors also descend as particle number increases and plateau when particle set reaches a certain size.

## 6. Conclusions and outlook

In this paper, a novel particle filter is developed for target tracking in the presence of glint noise. The filter transforms the construction of observation likelihood into an observation noise modeling problem. In the algorithm, non-Gaussian non-stationary noise is approximated by a Gaussian mixture model which is composed of a bank of weighted Gaussian components. The distribution parameters of each component are iteratively evaluated in the particle filter framework using MLE. The developed filter is capable of effectively describing the real noise density and obtaining a reliable state estimation in the presence of glint noise.

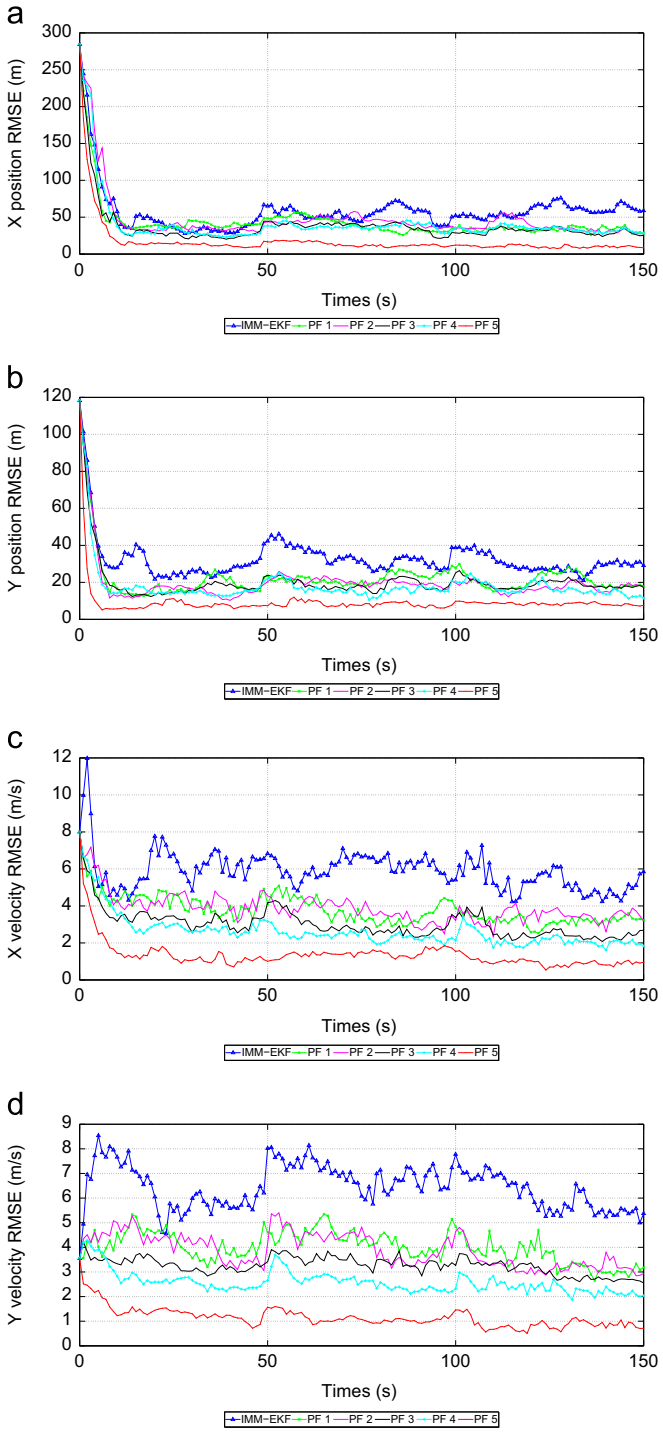


**Fig. 5.** Estimation errors of different filtering algorithms in maneuvering target tracking: (a) *X* position RMSE, (b) *Y* position RMSE, (c) *X* velocity RMSE, and (d) *Y* velocity RMSE.

To examine the performance of the new filter, bearings-only tracking simulations for both non-maneuvering and maneuvering targets were carried out using a simulation noise sequence sampled from a real glint noise record. For the purposes of comparison, several filtering algorithms were used in the simulations. The simulation results show that the proposed filter performs much better than comparable filters in both non-maneuvering and maneuvering target tracking problems, demonstrating high accuracy, fast convergence rate, and robustness.

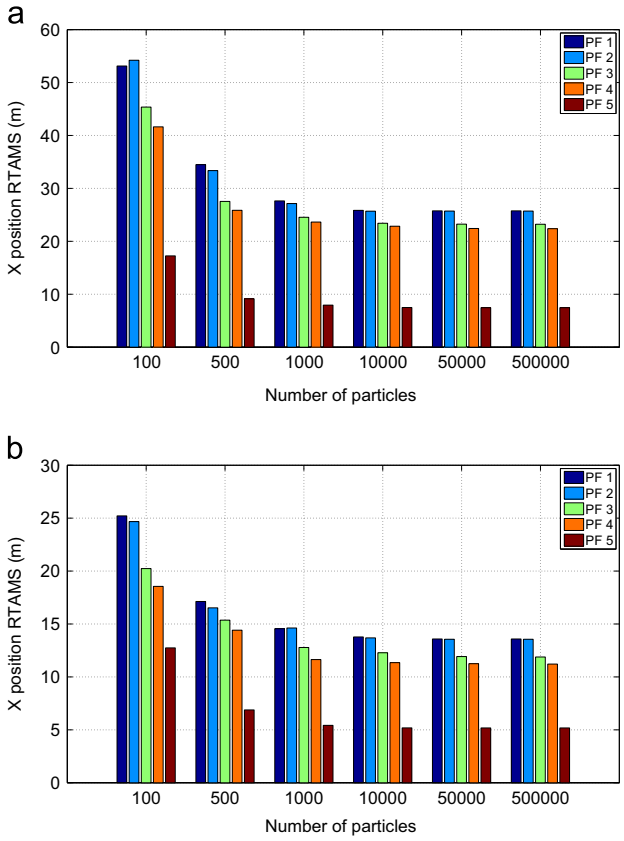


**Fig. 6.** Position estimation RTAMS of different PF algorithms with different particle numbers in maneuvering target tracking: (a) *X* position RTAMS and (b) *Y* position RTAMS.

**Table 4**
Performances of different PF algorithms in maneuvering tracking simulations with 100 particles.

| Filtering algorithm | *X* position RTAMS (m) | *Y* position RTAMS (m) | *X* velocity RTAMS (m/s) | *Y* velocity RTAMS (m/s) | Execution time (s) |
|---|---|---|---|---|---|
| PF 1 | 53.1243 | 25.2163 | 3.8575 | 4.0590 | 0.4635 |
| PF 2 | 54.2135 | 24.6729 | 3.9701 | 3.9376 | 0.4674 |
| PF 3 | 45.3654 | 20.2368 | 3.6354 | 3.5427 | 0.7453 |
| PF 4 | 41.6325 | 18.5536 | 3.3425 | 2.8461 | 0.9724 |
| PF 5(5) | 17.2456 | 12.7452 | 1.7694 | 1.2147 | 1.1533 |
| PF 5(10) | 16.7824 | 12.2846 | 1.7382 | 1.2024 | 3.4265 |
| PF 5(20) | 16.5376 | 12.1436 | 1.7256 | 1.1963 | 8.7316 |
| PF 5(100) | 16.5218 | 12.1455 | 1.7254 | 1.1965 | 75.4128 |

The proposed filter is designed to cope with a glint noise in target tracking system. In fact, it is a generic estimator for any kind of noise with Gaussian or non-Gaussian, stationary or non-stationary, zero mean or non-zero mean distribution. In other words, it can be used as a general means for effectively estimating the states of non-linear systems with noise. In regard to follow-up research, we plan to focus on establishing a general parameter initialization principle for Gaussian components in the noise mixture model that can be applied to various observation noises. Additionally, although increasing particle number improves the filter's accuracy, it simultaneously introduces a large computational load. As a result, it is worth discussing how to balance the estimation accuracy with the computational complexity of our filter in depth.

**Table 5**
Estimation RTAMS of the new PF algorithm with different numbers of particles and Gaussian components in maneuvering target tracking simulations.

| Number of particles | Number of Gaussian components | *X* position RTAMS (m) | *Y* position RTAMS (m) | *X* velocity RTAMS (m/s) | *Y* velocity RTAMS (m/s) |
|---|---|---|---|---|---|
| 100 | 5 | 17.2456 | 12.7452 | 1.7694 | 1.2147 |
| 100 | 10 | 16.7824 | 12.2846 | 1.7382 | 1.2024 |
| 100 | 20 | 16.5376 | 12.1436 | 1.7256 | 1.1963 |
| 100 | 100 | 16.5218 | 12.1455 | 1.7254 | 1.1965 |
| 200 | 5 | 13.8657 | 10.7524 | 1.4273 | 0.8633 |
| 200 | 10 | 13.7862 | 10.7154 | 1.4236 | 0.8628 |
| 200 | 20 | 13.7846 | 10.6749 | 1.4233 | 0.8622 |
| 200 | 100 | 13.7837 | 10.6735 | 1.4241 | 0.8614 |
| 500 | 5 | 9.1354 | 6.8835 | 0.8169 | 0.7241 |
| 500 | 10 | 9.1233 | 6.8457 | 0.8087 | 0.7223 |
| 500 | 20 | 9.1215 | 6.8246 | 0.8058 | 0.7215 |
| 500 | 100 | 9.1221 | 6.8243 | 0.8052 | 0.7217 |
| 1000 | 5 | 7.9462 | 5.4216 | 0.6962 | 0.6748 |
| 1000 | 10 | 7.9424 | 5.4162 | 0.6637 | 0.6741 |
| 1000 | 20 | 7.9368 | 5.4148 | 0.6631 | 0.6732 |
| 1000 | 100 | 7.9364 | 5.4151 | 0.6628 | 0.6728 |

## Acknowledgments

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions.

## References

[1] T. Rui, Q. Zhang, Y. Zhou, J. Xing, Object tracking using particle filter in the wavelet subspace, Neurocomputing 119 (2013) 125–130.
[2] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, Wiley, New York, 2006.
[3] O. Cappe, S.J. Godsill, E. Moulines, An overview of existing methods and recent advances in sequential monte carlo, Proc. IEEE 95 (5) (2007) 899–924.
[4] M.S. Grewal, A.P. Andrews, Kalman Filtering, 2nd ed., Wiley, New York, 2001.
[5] Z. Chen, Bayesian filtering: from Kalman filters to particle filters, and beyond, Statistics 182 (1) (2003) 1–69.
[6] A.J. Haug, A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and No-Gaussian Processes, Technical Report, MITRE Corporation, Mclean, 2005.
[7] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo methods for Bayesian filtering, Stat. Comput. 10 (3) (2000) 197–208.
[8] Y. Xia, Z. Deng, L. Li, X. Geng, A new continuous-discrete particle filter for continuous-discrete nonlinear systems, Inf. Sci. 242 (1) (2013) 64–75.
[9] F. Wang, Y.C. Sun, F.Y. Zhang, P. Liu, H.B. Min, Central difference particle filter applied to transfer alignment for sins on missiles, IEEE Trans. Aerosp. Electron. Syst. 48 (1) (2012) 375–387.
[10] J. MacCormick, A. Blake, A probabilistic exclusion principle for tracking multiple objects, Int. J. Comput. Vis. 39 (1) (2000) 57–71.
[11] F. Li, F. Qi, M. Shi, G.L. Zhang, Optimization-based particle filter for state and parameter estimation, J. Syst. Eng. Electron. 20 (3) (2009) 479–484.
[12] D. Hol, J.B. Schon, T.F. Gustafsson, On resampling algorithms for particle filters, in: IEEE Nonlinear Statistical Signal Processing Workshop, 2006.
[13] A.M. Johansen, A. Doucet, A note on auxiliary particle filters, Stat. Probab. Lett. 78 (12) (2008) 1498–1504.
[14] W. Hassan, N. Bangalore, P. Birch, R. Young, C. Chatwin, An adaptive sample count particle filter, Comput. Vis. Image Underst. 116 (12) (2012) 1208–1222.
[15] T. Li, S. Sun, T. Sattar, Adapting sample size in particle filters through KLD-resampling, Electron. Lett. 49 (12) (2013) 740–742.
[16] M.S. Arulampalam, S. Maskell, N.J. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[17] F. Septier, S.K. Pang, A. Carmi, S. Godsill, On mcmc-based particle methods for bayesian filtering: application to multitarget tracking, in: The Third IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2009.
[18] A. Giremus, J.Y. Tourneret, P.M. Djuric, An improved regularized particle filter for GPS/INS integration, in: IEEE Sixth Workshop on Signal Processing Advances in Wireless Communications, 2005.
[19] X.Y. Fu, Y.M. Jia, An improvement on resampling algorithm of particle filters, IEEE Trans. Signal Process. 58 (10) (2010) 5414–5420.
[20] F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.J. Nordlund, Particle filters for positioning, navigation, and tracking, IEEE Trans. Signal Process. 50 (2) (2002) 425–437.
[21] P. Li, R. Goodall, V. Kadirkamanathan, Estimation of parameters in a linear state space model using a Rao-Blackwellised particle filter, IEE Proc. Control Theory Appl. 151 (6) (2004) 727–738.
[22] J.H. Kotecha, P.M. Djuric, Gaussian sum particle filter, IEEE Trans. Signal Process. 51 (10) (2009) 2602–2612.
[23] K. Uosaki, T. Hatanak, State estimation of nonlinear stochastic systems by evolution strategies based Gaussian sum particle filter, in: Proceedings of the International Conference on Control, Automation and Systems, 2007.
[24] M.H. Zhang, M. Xin, J. Yang, Adaptive multi-cue based particle swarm optimization guided particle filter tracking in infrared videos, Neurocomputing 122 (25) (2013) 163–171.
[25] T.C. Li, S.D. Sun, T.P. Sattar, J.M. Corchado, Fight sample degeneracy and impoverishment in particle filters: a review of intelligent approaches, Expert Syst. Appl. 41 (8) (2014) 3944–3954.
[26] M. Bolic, P.M. Djuric, S.J. Hong, Resampling algorithms and architectures for distributed particle filters, IEEE Trans. Signal Process. 53 (7) (2005) 2442–2450.
[27] A. Mohammadi, A. Asif, A constraint sufficient statistics based distributed particle filter for bearing only tracking, in: The IEEE International Conference on Communications, 2012.
[28] J. Kim, M. Tandale, P. Menon, Particle filter for ballistic target tracking with glint noise, J. Guid. Control Dyn. 33 (6) (2010) 1918–1921.
[29] G.A. Hewer, R.D. Martin, J. Zeh, Robust preprocessing for kalman filtering of glint noise, IEEE Trans. Aerosp. Electron. Syst. 23 (1) (1987) 120–128.
[30] B.H. Borden, M.L. Mumford, A statistical glint/radar cross section target model, IEEE Trans. Aerosp. Electron. Syst. 19 (5) (1995) 781–785.
[31] W. Li, Y. Jia, J. Du, Z. Jun, Phd filter for multi-target tracking with glint noise, Signal Process. 94 (2014) 48–56.
[32] W. Wu, Target tracking with glint noise, IEEE Trans. Aerosp. Electron. Syst. 29 (1) (1993) 174–185.
[33] J.H. Kotecha, P.M. Djuric, Gaussian particle filter, IEEE Trans. Signal Process. 51 (10) (2003) 2593–2602.
[34] A. Doucet, N.J. Gordon, V. Krishnamurthy, Particle filters for state estimation of jump markov linear systems, IEEE Trans. Signal Process. 49 (3) (2001) 613–624.
[35] Z. Chen, Bayesian filtering: from Kalman filters to particle filters, and beyond (Ph.D. thesis), Mcmaster University, 2003.
[36] J. Gao, Z. Wei, Y. Meng, Z. WU, Particle filter based on observation likelihood importance sampling, J. Syst. Simul. 21 (12) (2009) 3705–3709.
[37] P.S. Maybeck, Stochastic Models, Estimation and Control, Academic Press, New York, 1982.
[38] E.A. Ibatoulline, Parameter estimation of non-Gaussian probability density of one-dimensional signals and interferences by iterative method of maximum likelihood, in: Proceedings of the 2003 IEEE International Symposium on Electromagnetic Compatibility, 2003.
[39] R. Karlsson, T. Schon, F. Gustafsson, Complexity analysis of the marginalized particle filter, IEEE Trans. Signal Process. 53 (11) (2005) 4408–4411.
[40] B. Ristic, S.M. Arulampalam, Tracking a manoeuvring target using angle-only measurements: algorithms and performance, Signal Process. 83 (2003) 1223–1238.
[41] A. Mukherjee, A. Sengupta, Likelihood function modeling of particle filter in presence of non-stationary non-Gaussian measurement noise, Signal Process. 90 (6) (2010) 1873–1885.
[42] H.W. Li, X.J. Ji, G.Q. Zhao, Toa-based target tracking using improved particle filter in passive bistatic radar with glint noise, in: Proceedings of the 2013 Sixth International Congress on Image and Signal Processing (CISP), 2013.
[43] X.H. Zhong, V.N. Hari, A.B. Premkumar, A.S. Madhukumar, Particle filtering with enhanced likelihood model for underwater acoustic source doa tracking, in: Proceedings of the 2011 IEEE—Spain OCEANS, 2011.
[44] E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan, Interacting multiple model methods in target tracking: a survey, IEEE Trans. Aerosp. Electron. Syst. 34 (1) (1998) 103–123.
[45] Y. Boers, J.N. Driessen, Interacting multiple model particle filter, IEE Proc. Radar Sonar Navig. 150 (5) (2013) 344–349.
[46] H.W. Li, J. Wang, Particle filter for manoeuvring target tracking via passive radar measurements with glint noise, IET Radar Sonar Navig. 6 (3) (2012) 180–189.

**Hangyuan Du** received the B.S. degree from Heilongjiang Institute of Technology, in 2008, and the Ph.D. degree in Control Science and Engineering from Harbin Engineering University, in 2012. He is currently a member of School of Computer and Information Technology at Shanxi University, as a lecturer. His research interests lie in the areas of data fusing, nonlinear filter and mobile robotics.

**Wenjian Wang** is a Professor in the School of Computer and Information Technology and Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education at Shanxi University. She received the Ph.D. degree from Xi'an Jiaotong University, in 2004. Her current research interests include computational intelligence, support vector machine, data mining and knowledge discovery.

**Liang Bai** studied computer science at the Shanxi University, and received his Ph.D. degree, in June 2012. He is an Assistant Professor in the School of Computer and Information Technology and Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education at Shanxi University. His areas of research include data fusing, rough set and machine learning.