RESEARCH

Open Access

Confidence partitioning sampling filtering



Xingzi Qiang¹, Rui Xue^{2*} and Yanbo Zhu³

*Correspondence: xuerui@buaa.edu.cn

 ¹ School of Internet, Anhui University, Hefei 230039, People's Republic of China
 ² School of Electronics and Information Engineering, Beihang University, Beijing 100191, People's Republic of China
 ³ The Aviation Data
 Communication Corporation, Floor 16, Bai Yan Building, No.
 238, Bei Si Huan Zhong Road, Haidian District, Beijing 100191, People's Republic of China

Abstract

The confidence partitioning sampling filter (CPSF) method proposed in this paper is a novel approach for solving the generic nonlinear filtering problem. First, the confidence probability space (CPS) is defined, which restricts the state transition in a bounded and closed state space in the recursive Bayesian filtering. In the posterior CPS, the weighted grid samples, represented the posterior PDF, are obtained by using the partitioning sampling technique (PST). Each weighted grid sample is treated as an impulse function. The approximate expression of the posterior PDF, as key for the PST implementation, is obtained by using the properties of the impulse function in the integral operation. By executing the selection of the CPS and the PST step repeatedly, the CPSF framework is formed to achieve the approximation of the recursive Bayesian filtering. Second, the difficulty of the CPSF framework implementation lies in obtaining the real posterior CPS. Therefore, the space intersection (SI) method is suggested to obtain the approximate posterior CPS. On this basis, the SI CPSF algorithm, as an executable algorithm, is formed to solve the generic nonlinear filtering problem. Third, the approximate error between the CPSF framework and the recursive Bayesian filter is analyzed theoretically. The consistency of the CPSF framework to the recursive Bavesian filter is proved. Finally, the performances of the SI CPSF algorithm, including robustness, accuracy and efficiency, are evaluated using four representative simulation experiments. The simulation results showed that SI CSPF requires far less samples than particle filter (PF) under the same accuracy. Its computation is on average one order of magnitude less than that of the PF. The robustness of the proposed algorithm is also evaluated in the simulations.

Keywords: Confidence, Nonlinear filtering, Partitioning integral, Posterior distribution, State estimation

1 Introduction

It is well known that Bayesian filtering provides a general recursive solution to the filtering problem by inferring the posterior probability density function (PDF) of the state space model with hidden states [1-5]. For a linear Gaussian system, the well-known Kalman filter (KF) can efficiently and accurately obtain the analytical solution of Bayesian inference [6-8]. However, it is impossible to obtain the analytical solution of Bayesian inference using the transfer of finite parameters in a nonlinear/non-Gaussian system. Therefore, several approximate methods were proposed to solve the posterior PDF and estimate the state in engineering applications [9-12].



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativeCommons.org/licenses/by/4.0/.

According to the approximation principles, the existing nonlinear filtering methods can be grouped into local inference and global inference [13, 14]. These approaches usually revolve around two core problems: (1) How to accurately describe the posterior PDF at current step for obtaining an estimation closer to the Bayesian inference? (2) How to accurately transfer the posterior PDF from the current to the next step for ensuring the continuous execution of recursive Bayesian inference?

For local inference, the typical representative is the Kalman-like filters including extended KF (EKF) [15–17], unscented KF (UKF) [18, 19], cubature KF (CKF) [20, 21], quadrature KF (QKF) [13, 22], etc. All of these methods use Gaussian distributions to approximate the real posterior PDF for solving the core problem (1). Their difference lies in solving the core problem (2). The EKF linearizes a nonlinear system with a firstorder Taylor series expansion, so that the posterior PDF is transferred in the linear part of a nonlinear system. The UKF describes a Gaussian distribution using obtained 2d + 1sigma points before transferring the posterior PDF, where d denotes the dimensions of the state. The third-degree spherical-radial cubature rule is used to overcome the uncertainty of a nonlinear system in the CKF. Therefore, the calculation of volume point set becomes the key of the transfer of the posterior PDF. Similarly, the quadrature rule is used to obtain the sigma point set and transfer the posterior PDF in the QKF. The [14, 23] approximate the posterior PDF using a multiparameter mixed distribution. The resulting filter is referred to as the belief condensation filtering (BCF). In this method, the distribution model is preset. The optimization algorithm, based on the Kullback-Leibler divergence [24], is used to calculate the parameters of the distribution. All the above methods use finite parameter distribution to describe the real posterior PDF.

For global inference, a typical representative is the sequential Monte Carlo (SMC)based filtering methods [16, 25, 26]. As the major directions of SMC method, particle filter (PF) uses many weighted particles to describe the posterior PDF and simulate the propagation characteristics of the probability distribution [27, 28]. Weighted particles can flexibly describe arbitrary distribution, which makes PF have a greater potential to solve complex nonlinear/non-Gaussian filtering problems [29, 31]. However, the development of the PF is always confused by the particle degeneracy, impoverishment problems and dimension curse, which might affect the accuracy and efficiency and even lead to filtering divergence [32-36]. In addition, computational complexity is a crucial aspect of concern in PF, and it has the potential to significantly restrict the applicability of nonlinear filtering methods [37, 38]. Since the advantages and the disadvantages of PF, researchers have proposed several techniques to enhance the stability, accuracy and efficiency of the PF, which mainly include the resampling technique [39-44], the improvement of the proposed distribution [45-47] and the integration technique [48, 49]. For mixed linear/nonlinear state space models, Rao-Blackwellized PF (RBPF) uses KF and the PF to achieve the state estimation of the linear and nonlinear parts, respectively. This method can alleviate the dimension curse suffered by PF [11, 50]. Furthermore, the stratification technique is used in [51], to compress the statistical information contained in a set of random particles. The number of particles can be compressed by the method according to the probability of different regions, so as to improve the filtering efficiency. Given the above, the idea of PF-like methods is that the high probability region of the posterior PDF dominates the performance of filtering. Therefore, the improvement of PF-like methods is essentially to find a way to improve the efficiency of particle for describing the posterior PDF. Another typical representative of global inference is the grid-based filters (GF) [9, 10]. GF is another optimal filtering method compared with the KF. It is worth noting that this applies only when the state space is discrete and consists of a finite number of states. In this case, the integrals of the recursive Bayesian filtering can be executed numerically as a discrete, nonlinear convolution [52-54]. However, the application scope of GF is quite limited. Therefore, GF has not been paid attention by most scholars for a long time.

Generally speaking, local inference approximates the posterior PDF using one or more mixed known distribution. Its advantage is that the calculation cost is small, so it has high efficiency. For global inference, SMC method approximates the posterior PDF by accumulating a large number of weighted particles. The posterior PDF is transferred through Monte Carlo method. They require much more computation than local inference methods. GF method uses numerical integration to describe and transfer the posterior PDF. It has strict constraint for the system environment. However, the global inference methods have better accuracy and robustness when dealing with some special nonlinear filtering problems.

In this paper, a novel non-parameter filter method called the confidence partitioning sampling filtering (CPSF) method is proposed. We have absorbed the idea of focusing on the high probability regions in the SMC method. On this basis, the concept of the confidence probability space (CPS) is defined, which is used to choose the high probability region of the posterior PDF as the posterior CPS. Then, the grid samples are generated in the posterior CPS. The posterior probability of each grid sample, also called the weight of sample, is calculated according to the approximate expression of the posterior PDF, where the acquisition of the approximate expression is based on the idea of the numerical integration in GF and the method of the transfer of the posterior PDF at that time. The transfer of the posterior PDF is executed around obtaining the approximate expression. The main contributions arising from this study are as follows:

- (1) The CPSF framework is established as an idea to solve the generic nonlinear filter problem in theory. First, the concept of the CPS is defined to compress the state space under a certain confidence condition maximally. Second, the partitioning sampling technique (PST) is performed in the posterior CPS, to obtain some weighted grid samples for describing the posterior PDF. The properties of the impulse function in the integral operation guide us to obtain the approximate expression of the posterior PDF, which is used in the process of the PST. Finally, the CPSF framework is formed by selecting the posterior CPS and executing the PST repeatedly, which approximates the Bayesian inference.
- (2) The SI_CPSF algorithm is proposed as a specific nonlinear filtering algorithm. To overcome the difficulty of obtaining the real posterior CPS in the CPSF framework, the space intersection (SI) method is proposed to obtain the approximate posterior CPS. Then, an executable nonlinear filtering algorithm, called SI_CPSF algorithm, is formed to solve the generic nonlinear filter problems.

- (3) The approximate error between the CPSF framework and the recursive Bayesian filter is analyzed theoretically. The approximation error is divided into two parts: the space truncation error and quantization error. These two types of error can be controlled by two parameters in the CPSF framework: confidence and sampling interval. Furthermore, the consistency of the CPSF framework to the Bayesian filter is proved theoretically.
- (4) The performances of the proposed algorithm, including the accuracy, efficiency and robustness, are evaluated by four representative simulation experiments.

The structure of this paper is as follows: In Sect. 2, the principal problem model and Bayesian filtering are introduced. In Sect. 3, the CPSF framework and the SI_CPSF algorithm which is derived from the SI technique is presented in detail. In Sect. 4, the approximate error of the CPSF framework to the Bayesian filter is discussed at great length. Furthermore, the consistency of the CPSF framework to Bayesian filter is proved. In Sect. 5, simulation results are provided that describe the performance of the SI_CPSF algorithm. In Sect. 6, this paper is concluded.

2 Problem statement

The discrete state space model of the general filtering problem at time step k can be described as follows:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{u}_k \tag{1}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where \mathbf{x}_k and \mathbf{y}_k are the state and observation, respectively. \mathbf{u}_k and \mathbf{v}_k are noise sequences with $\mathbf{u}_k \sim p_{\mathbf{u}}(\mathbf{x})$ and $\mathbf{v}_k \sim p_{\mathbf{v}}(\mathbf{y})$. They are independent of each other. \mathbf{f} and \mathbf{h} denote the state model and observation model, respectively. Alternatively, the complete filter process can be described as the transition of probability distribution: State transition distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ can be obtained from (1) and $p_{\mathbf{u}}(\mathbf{x})$; the likelihood distribution $p(\mathbf{y}_k | \mathbf{x}_k)$ can be obtained from (2) and $p_{\mathbf{v}}(\mathbf{y})$.

The objective of recursive Bayesian filtering is to obtain the posterior PDF at the current time step, to achieve the state estimation [42]. Recursive Bayesian filtering which can be divided into two steps: *Prediction and Update*.

The prediction step infers the prior distribution at time step *k*, as follows:

$$p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_{k}|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}$$
(3)

The update infers the posterior distribution at time step *k* as follows:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}$$
(4)

3 Confidence partitioning sampling filtering

In this section, the CPSF method is proposed to overcome the integral difficulty that is caused by a nonlinear system in the filtering process. The main idea of the method is that the high probability region of the posterior PDF dominates the performance of the filtering. Therefore, the whole structure of the CPSF method is finding the high probability region and describing the posterior PDF in the region.

3.1 Partitioning sampling technique

Definition 1 Suppose that $p(\mathbf{x})$ is the PDF of a distribution. Its complete probability space is called $\mathbb{C}_{p(\mathbf{x})}^{0}$, $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ is a bounded subspace of it, which satisfies

minimize S

s.t
$$\int_{C_{p(\mathbf{x})}^{\alpha}} p(\mathbf{x}) d\mathbf{x} = 1 - \alpha$$
 (6)

where $S = \int_{\mathbb{C}_{p(\mathbf{x})}^{\alpha}} \mathbf{1} d\mathbf{x}$, $0 \le \alpha \le 1$. Then, $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ is called the CPS of $p(\mathbf{x})$ under the confidence $1 - \alpha$.

In the $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$, the sampling interval $\boldsymbol{\tau}$ is used to divide it into L blocks, where $\boldsymbol{\tau} = [\tau_1, \tau_2, \cdots, \tau_D]^T$. D is the dimension. Then, samples are deployed on the center of each block and obtain the sample set $\hat{\mathbf{X}}$, where $\hat{\mathbf{X}} \equiv [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \cdots, \hat{\mathbf{x}}_L]^T$. The weight of each sample is calculated as follows:

$$\boldsymbol{\omega} \equiv \left[\omega_1, \omega_2, \cdots \omega_L\right]^T \tag{7}$$

$$\omega_l = \frac{p(\hat{\mathbf{x}}_l)}{\sum_{l=1}^L p(\hat{\mathbf{x}}_l)} \tag{8}$$

The above process is called partitioning sampling technique (PST). The detailed steps are shown in Table 1

Theorem 1 Suppose that $p(\mathbf{x})$ is the PDF of a distribution. $\{\hat{\mathbf{X}}, \boldsymbol{\omega}\}_L$ is obtained using the PST from $p(\mathbf{x})$. Then,

 Table 1
 Partitioning sampling technique

```
INPUT: p(\mathbf{x}), \mathbb{C}_{p(\mathbf{x})}^{\alpha} \mathbf{\tau}

OUTPUT: \{ \hat{\mathbf{X}}, \boldsymbol{\omega} \}_{L}

Procedure:

1): Sampling \hat{\mathbf{X}} in \mathbb{C}_{p(\mathbf{x})}^{\alpha} according to \mathbf{\tau};

2): Calculate weights according to (7), (8) and get \{ \hat{\mathbf{X}}, \boldsymbol{\omega} \}_{L}
```

$$p(\mathbf{x}) \propto \lim_{\alpha, \tau \to \mathbf{0}} \sum_{l=1}^{+\infty} \omega_l \delta\left(\mathbf{x} - \hat{\mathbf{x}}_l\right)$$

Proof See Appendix A.

For instance, suppose $p(\mathbf{x}) \sim N(\mathbf{0}, diag([1, 1]))$. The $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ is a circle of equal probability in two-dimensional space. Let $\alpha = 0.001$, the radius of the circle is:

$$r = \sqrt{-2\ln(\alpha)} \approx 3.717$$

Let $\boldsymbol{\tau} = [0.5, 0.5]^T$, the processing procedure of the PST can be seen in Fig. 1 intuitively. Finally, $\{\hat{\mathbf{X}}, \boldsymbol{\omega}\}_{\tau}$ can be used to approximately express the $p(\mathbf{x})$ discretely:

$$p(\mathbf{x}) \propto \sum_{l=1}^{L} \omega_l \delta\left(\mathbf{x} - \hat{\mathbf{x}}_l\right)$$
(9)

where $\delta(\cdot)$ denotes the Dirichlet function.

In this instance, the subspace $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ contains $(1 - \alpha) \times 100\%$ probability information. Simultaneously, $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ is the minimum space under the condition of ignoring $\alpha \times 100\%$ probability information. In a general probability model (e.g., Gaussian distribution, mixture Gaussian distribution, etc.), the probability space can be greatly compressed into a bounded probability space $\mathbb{C}_{p(\mathbf{x})}^{\alpha}$ with losing a negligible amount of probability information. This makes it possible to implement the PST [55, 56].

3.2 Confidence partitioning sampling filtering framework

In this subsection, we illustrate the CPSF framework using the Bayesian inference from time step k - 1 to k. Suppose that the posterior PDF at time step k - 1 can be described by a set of weighted grid samples $\{\hat{\mathbf{X}}_{k-1}, \boldsymbol{\omega}_{k-1}\}_{N_{k-1}}$, where N_{k-1} denotes the sample number. $\hat{\mathbf{X}}_{k-1} \equiv [\hat{\mathbf{x}}_{k-1,1}, \hat{\mathbf{x}}_{k-1,2}, \dots, \hat{\mathbf{x}}_{k-1,N_{k-1}}]^T$ deployed in the bounded space $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ evenly. Therefore, all the probability information is described by their weights.

Then, the posterior PDF at time step k - 1 can be expressed discretely as follows:

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \propto \sum_{n=1}^{N_{k-1}} \omega_{k-1,n} \delta(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1,n})$$
(10)



In (10), the posterior PDF at time step k - 1 is expressed by N_{k-1} weighted impulse functions. Then, the prior PDF at time step k can be considered as a result that the impulse functions act on $\mathbf{f}(\cdot)$. The partial prior PDF, which is caused by one impulse function, can be expressed according to (3) as follows:

$$p_n(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \propto \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\omega_{k-1,n}\delta(\mathbf{x}_{k-1}-\hat{\mathbf{x}}_{k-1,n})d\mathbf{x}_{k-1}$$
(11)

Meanwhile, (11) can be rewritten according to the sampling property of the impulse function:

$$p_n(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \propto \omega_{k-1,n} p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1,n})$$
(12)

Furthermore, $p(\mathbf{x}_k | \hat{\mathbf{x}}_{k-1,n})$ is translated $p_{\mathbf{u}}(\mathbf{x})$ by adding $\vec{\mathbf{x}}_{k,n}$. Concretely, it can be expressed as follows:

$$p(\mathbf{x}_k | \hat{\mathbf{x}}_{k-1,n}) = p_{\mathbf{u}}(\mathbf{x}_k - \vec{\mathbf{x}}_{k,n})$$
(13)

where $\vec{\mathbf{x}}_{k,n} = \mathbf{f}(\hat{\mathbf{x}}_{k-1,n})$. Substituting (12) into (13) yields:

$$p_n(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \propto \omega_{k-1,n} p_{\mathbf{u}}(\mathbf{x}_k - \vec{\mathbf{x}}_{k,n})$$

Now, the partial prior PDF is described by the PDF of process noise. Moreover, the complete prior PDF expression can be obtained according to the additive property of the impulse function, as follows:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \propto \sum_{n=1}^{N_{k-1}} \omega_{k-1,n} p_{\mathbf{u}}(\mathbf{x}_k - \vec{\mathbf{x}}_{k,n})$$
(14)

and the posterior PDF at time step k can be expressed according to (4) and (14) as follows:

$$p(\mathbf{x}_{k}|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_{k}|\mathbf{x}_{k}) \sum_{n=1}^{N_{k-1}} \omega_{k-1,n} p_{\mathbf{u}}(\mathbf{x}_{k} - \vec{\mathbf{x}}_{k,n})$$

$$\propto p_{\mathbf{v}}(\mathbf{y}_{k} - \mathbf{h}(\mathbf{x}_{k})) \sum_{n=1}^{N_{k-1}} \omega_{k-1,n} p_{\mathbf{u}}(\mathbf{x}_{k} - \vec{\mathbf{x}}_{k,n})$$
(15)

In (15), the approximate expression of the posterior PDF is constructed using the process noise model, the observation noise model and the weights of the grid samples at time step k-1. We can theoretically obtain the posterior CPS according to (15). Then, the PST can be executed for obtaining the weighted grid samples $\{\hat{\mathbf{X}}_k, \boldsymbol{\omega}_k\}_{N_k}$ at time step k. The posterior PDF at time step k can be described in discrete form, as follows:

$$p(\mathbf{x}_{k}|\mathbf{y}_{1:k}) \propto \sum_{n=1}^{N_{k-1}} \omega_{k,n} \delta(\mathbf{x}_{k} - \hat{\mathbf{x}}_{k,n})$$
(16)

The state at time step *k* can be estimated:

$$\bar{\mathbf{x}}_k = (\mathbf{\omega}_k)^T \hat{\mathbf{X}}_k \tag{17}$$



Fig. 2 Structure of the CPSF framework



Initialization: Set $\alpha, \mathbf{\tau}$ Calculate $\mathbb{C}_{p(\mathbf{x}_0)}^{\alpha}$ Obtain the weighted grid samples according to PST $\left\{ \hat{\mathbf{X}}_{0}, \boldsymbol{\omega}_{0} \right\}_{N_0} \leftarrow PST\left(p(\mathbf{x}_0), \mathbb{C}_{p(\mathbf{x}_0)}^{\alpha}, \mathbf{\tau} \right)$ //Overall time steps: For $k \leftarrow 1$ to K do 1): Get the expression of the posterior PDF at time step k (see Eq. 15); 2): Obtain the posterior CPS at time step k according to Definition 1 and Eq. (15); 3): Obtain the weighted grid samples according to PST $\left\{ \hat{\mathbf{X}}_{k}, \boldsymbol{\omega}_{k} \right\}_{N_k} \leftarrow PST\left(p(\mathbf{x}_k | \mathbf{y}_{1:k}), \mathbb{C}_{p(\mathbf{x}_k | \mathbf{y}_{1:k})}^{\alpha}, \mathbf{\tau} \right)$

```
4): State estimation (see Eq. 17)
End
```

After that, the recursive Bayesian filtering is approximately executed by obtaining the approximate expression of the posterior PDF, finding the posterior CPS and performing the PST, repeatedly.

This method, which approximates the Bayesian filtering, is called the CPSF framework. In this framework, we give two descriptive forms of the posterior PDF: the approximate expression (see (15)) and the weighted grid samples $\{\hat{\mathbf{X}}_k, \boldsymbol{\omega}_k\}_{N_k}$. The transfer of the posterior PDF is achieved by the repeated execution of the two steps: finding the posterior CPS and performing the PST. The structure of the CPSF framework is shown in Fig. 2. The filter steps are shown in detail in Table 2

Remark 1 Compared with the computation of the PF, the CPSF framework adds the calculation step of the sample weight, which is mainly reflected in (15). The computational complexity of this part is about $O(N_{k-1} \times N_k)$. However, samples do not need to reflect the probability information. This phenomenon greatly reduces the requirement of the sample number, which can improve the efficiency of filtering. The efficiency is evaluated using simulation experiments in Section V.

Remark 2 Generally speaking, the posterior PDF can obtain the better estimation than the prior PDF or likelihood PDF, which means it has better compressibility than that

of the prior or likelihood in most cases. For a relatively stationary filtering system, the size of the CPS is relatively stable in the whole filtering process. Therefore, the sample number obtained by the PST is relatively stable for a determined τ . For a non-stationary filtering system, the size of the CPS might be changed with the change of the scale of the posterior distribution. In this case, the samples number might fluctuate greatly in the whole filtering process for a determined τ . In this case, we can dynamically adjust the sampling interval according to the size of the posterior CPS.

Remark 3 In the CPSF framework, the selection of two parameters (α and τ) depends on the following principles: (i) The selection of α depends on the engineering requirements for the credibility of the filtering results. In specific engineering requirements, there is usually a strict demand for the confidence level of filtering results. For instance, in civil aviation, the ICAO GNSS SARPs specify a integrity requirement of 10^{-7} for CAT I. (ii) The selection of the sampling interval depends on the size of the confidence space for the filter to enter the steady-state working phase and the requirements for computational timeliness.

3.3 CPSF algorithm based on space intersection

In the previous subsection, we proposed the CPSF framework based on the concept of the CPS and the properties of the impulse function. For a simple distribution, the CPS can be obtained easily according to (15). For a complex nonlinear system, it is difficult to obtain the posterior CPS directly. To ensure the implementation of the CPSF framework in arbitrary nonlinear system, the SI technique is suggested for estimating the posterior CPS roughly in this subsection, which forms a nonlinear filter algorithm called SI_CPSF algorithm.

Suppose that the $\mathbb{C}_{p_u}^{\alpha}$ is the CPS of the process noise model. $\left\{\hat{\mathbf{X}}_{k-1}, \boldsymbol{\omega}_{k-1}\right\}_{N_{k-1}}$ is the weighted grid sample for describing the posterior PDF at time step k-1, which satisfies

$$\hat{\mathbf{X}}_{k-1} \in \mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$$

After one time step transfer of the system model, $\vec{\mathbf{X}}_k$ can be obtained, where

$$\vec{\mathbf{X}}_k = \begin{bmatrix} \vec{\mathbf{x}}_{k,1}, \ \vec{\mathbf{x}}_{k,2}, \dots, \ \vec{\mathbf{x}}_{k,N_{k-1}} \end{bmatrix}^T$$

Furthermore, each sample $\hat{\mathbf{x}}_{k-1,n}$ in $\hat{\mathbf{X}}_{k-1}$ will be transferred in the space $\vec{\mathbf{x}}_{k,n} + \mathbb{C}_{p_{\mathbf{u}}}^{\alpha}$ under the confidence $1 - \alpha$. Therefore, the estimation of the prior CPS can be described as follows:

$$\tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{\alpha} = \bigcup_{n=1}^{N_{k-1}} \left(\vec{\mathbf{x}}_{k,n} + \mathbb{C}_{p_{\mathbf{u}}}^{\alpha} \right)$$
(18)

The likelihood CPS is $\mathbb{C}_{p(\mathbf{y}_k|\mathbf{x}_k)}^{\alpha}$. As $p(\mathbf{y}_k|\mathbf{x}_k)$ is determined by the observation and its noise model, we can obtain the CPS of likelihood distribution expediently. The roughly estimation of the posterior CPS is defined as:

$$\tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})} = \tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})} \cap \mathbb{C}^{\alpha}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}$$
(19)

As the expression of the posterior PDF can be approximated by (15), we can perform the PST in the $\tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{\alpha}$. Then, the $\{\tilde{\mathbf{X}}_{k}, \tilde{\omega}_{k}\}_{\tilde{N}_{k}}$ can be obtained according to the set sampling interval.

Proposition 1 Suppose $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ and $\{\hat{\mathbf{X}}_{k-1}, \boldsymbol{\omega}_{k-1}\}_{N_{k-1}}$ are the real posterior CPS and the weighted grid samples at time step k-1, respectively. When the $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ is applied to arbitrary nonlinear system $\mathbf{f}(\cdot)$, the lower bound of the confidence of $\tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{\alpha}$ approximates to $1 - (3\alpha - \alpha^{2})$.

Proof See Appendix B.

For the estimation of the posterior CPS, we can determine its lower bound: $\mathbb{C}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{3\alpha-\alpha^{2}} \subset \tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{\alpha}$. In practice, the weak amplification of the lower bound of the confidence has little effect on the state estimation as α is very small. However, the change of the position relation between the likelihood CPS and the prior CPS might affect its upper bound, which makes that it impossible to determine its upper bound. This phenomenon might result in containing extremely low weighted samples in the weighted grid samples. These low weighted samples would greatly limit the efficiency of filtering. Fortunately, the probability information is completely reflected in the weights for the weighed grid samples. The samples are evenly distributed within the approximate posterior CPS, each sample represents only a subspace within it. Therefore, we take the weight as the evaluation index to discard the lowest weighted samples:

$$\left\{\hat{\mathbf{X}}_{k},\boldsymbol{\omega}_{k}\right\}_{N_{k}} \Leftarrow \left\{\tilde{\mathbf{X}}_{k},\tilde{\boldsymbol{\omega}}_{k}\right\}_{\tilde{N}_{k}}$$
(20)

In this procedure, we discard the \overline{N}_k (where, $N_k = \tilde{N}_k - \overline{N}_k$) samples with the lowest weights, which satisfies:

$$\sum_{n=1}^{\overline{N}_k} \tilde{\omega}_{k,n} \leq \alpha,$$

and normalize the weight of the rest samples to get $\boldsymbol{\omega}_k$. Then, the weighted grid samples set $\{\hat{\mathbf{X}}_k, \boldsymbol{\omega}_k\}_{N_k}$ is obtained, which roughly belongs to $\mathbb{C}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}^{\alpha}$. This specify filtering algorithm is called SI_CPSF algorithm. The block diagram of it is shown in Fig. 3. The detailed algorithm flow is shown in Table 3.

4 Approximate error of the CPSF method to Bayesian filter

In this section, the approximation error between the CPSF method and Bayesian filtering is analyzed theoretically. The theoretical consistency between the CPSF method and Bayesian filtering is further proved.

The filtering process from time step k - 1 to k is considered as an example: The posterior PDF at time step k - 1 is $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$. $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{0}$ is the complete state space including two parts: $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ and $\overline{\mathbb{C}}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$, where $\overline{\mathbb{C}}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ is the complementary set of $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$. Therefore:



Fig. 3 Block diagram of the SI_CPSF algorithm

Table 3 SI_CPSF algorithm

Initialization: Set $\alpha, \mathbf{\tau}$ Calculate $\mathbb{C}^{\alpha}_{\rho_{\mathbf{u}}(\mathbf{x})}\mathbb{C}^{\alpha}_{\rho(\mathbf{x}_{0})}$ Implement partitioning sampling technique $\{\hat{\mathbf{x}}_{0}, \boldsymbol{\omega}_{0}\} \leftarrow PST(\rho(\mathbf{x}_{0}), \mathbb{C}^{\alpha}_{e_{i}}, \mathbf{\tau})$

For $k \leftarrow 1$ to K do 1): Get the expression of the posterior PDF at time step k (see Eq. 15); 2): Get the estimation of the prior CPS (see Eq. 18); 3): Get the estimation of the posterior CPS (see Eq. 19); 4): Implement partitioning sampling technique: $\{\tilde{\mathbf{X}}_{k}, \tilde{\omega}_{k}\}_{z} \leftarrow PST(p(\mathbf{x}_{k}|\mathbf{y}_{1:k}), \tilde{\mathbb{C}}_{n(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{z}, \mathbf{\tau})$

$$\{\hat{\mathbf{x}}_{k}, \boldsymbol{\omega}_{k}\}$$

5): Space contraction: Obtain ^k, ^w ^k _{Nk} according to (20)
6): State estimation (see Eq. 17)
end

$$\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha} \cap \overline{\mathbb{C}}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha} = \emptyset$$

$$\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha} \cup \overline{\mathbb{C}}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha} = \mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{0}$$

Others,

$$\int_{\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}} p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} = 1 - \alpha$$

$$\int_{\overline{\mathbb{C}}_{p}^{\alpha}(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})} p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} = \alpha$$

 ε is defined as the approximation error between the CPSF method and Bayesian filter, as follows:

$$\varepsilon = \mathbf{x}_{k,bayes} - \sum_{i=1}^{N_k} \omega_{k,i} \mathbf{x}_{k,i}$$
(21)

where $\mathbf{x}_{k, bayes}$ denotes the Bayesian estimation at time step *k*.

For convenience, we define: $p_k^{li} = p(\mathbf{x}_k | \mathbf{y}_{1:k}), p_k^{tr} = p(\mathbf{x}_k | \mathbf{x}_{k-1}), p_{k-1}^{po} = p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}), p_{k-1}^{pr} = p(\mathbf{x}_k | \mathbf{y}_{1:k-1}).$

Then,

$$\mathbf{x}_{k,bayes} = \int_{\mathbb{C}_{p_{k}^{o}}^{0}} \mathbf{x}_{k} p_{k}^{po} d\mathbf{x}_{k}$$

$$= \int_{\mathbb{C}_{p_{k}^{o}}^{\alpha}} \mathbf{x}_{k} p_{k}^{po} d\mathbf{x}_{k} + \int_{\mathbb{C}_{p_{k}^{o}}^{\alpha}} \mathbf{x}_{k} p_{k}^{po} d\mathbf{x}_{k}$$

$$= \int_{\mathbb{C}_{p_{k}^{o}}^{\alpha}} \mathbf{x}_{k} p_{k}^{po} d\mathbf{x}_{k} / (1 - \alpha) + \mathbf{\xi}_{1}$$
(22)

where

$$\boldsymbol{\xi}_{1} = \int_{\overline{\mathbb{C}}_{p_{k}^{p_{o}}}} \mathbf{x}_{k} p_{k}^{p_{o}} d\mathbf{x}_{k} - \frac{\alpha}{1-\alpha} \int_{\mathbb{C}_{p_{o}^{p_{o}}}} \mathbf{x}_{k} p_{k}^{p_{o}} d\mathbf{x}_{k}$$
(23)

In (22), the first item is considered as the approximate expectation of $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ in $\mathbb{C}_{p_k}^{\alpha_{po}}$. The second item of (22) is considered as a truncation error. In (23), the error of the first item is due to the absence of the $\overline{\mathbb{C}}_{p_k}^{\alpha_{po}}$. The second term of (23) is the compensation of the first item due to the normalization of the posterior PDF at time step *k* in the $\mathbb{C}_{p_k}^{\alpha_{po}}$. After introducing (3) and (4) into the first term of (22), we have:

$$\begin{split} &\int_{\mathbb{C}_{p_{k}^{p_{p}}}^{\alpha}} \frac{\mathbf{x}_{k} p_{k}^{p_{p}}}{1-\alpha} d\mathbf{x}_{k} \\ &= \frac{\int_{\mathbb{C}_{p_{k}^{p_{p}}}^{\alpha}} \mathbf{x}_{k} p_{k}^{li} \int_{\mathbb{C}_{p_{p}^{p}}^{0}} p_{k}^{tr} p_{k-1}^{p_{0}} d\mathbf{x}_{k-1} d\mathbf{x}_{k}}{(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1})} \\ &= \frac{\int_{\mathbb{C}_{p_{k}^{p_{p}}}^{\alpha}} \mathbf{x}_{k} p_{k}^{li} \left(\int_{\mathbb{C}_{p_{k}^{p_{p}}}^{\alpha}} p_{k}^{tr} p_{k-1}^{p_{0}} d\mathbf{x}_{k-1} + \int_{\mathbb{C}_{p_{k}^{p_{p}}}^{\alpha}} p_{k}^{tr} p_{k-1}^{p_{0}} d\mathbf{x}_{k-1} \right) d\mathbf{x}_{k}}{(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1})} \end{split}$$

$$(24)$$

$$&= \int_{\mathbb{C}_{p_{k}^{\alpha}}}^{\alpha} \mathbf{x}_{k} p_{k}^{li} A d\mathbf{x}_{k} \bigg/ \left[(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1}) \right] + \mathbf{\xi}_{2}$$

where

$$A = \int_{\mathbb{C}_{p_k^{pr}}} p_k^{tr} p_{k-1}^{po} d\mathbf{x}_{k-1} / (1-\alpha)$$
(25)

$$\boldsymbol{\xi}_{2} = \int_{\substack{\mathbb{C}^{\alpha}_{po}\\p_{k}^{\prime}}} \mathbf{x}_{k} p_{k}^{li} B d\mathbf{x}_{k} / \left[(1-\alpha) p \left(\mathbf{y}_{k} \middle| \mathbf{y}_{1:k-1} \right) \right]$$
(26)

$$B = \int_{\mathbb{C}_{p_k}^{\alpha}} p_k^{tr} p_{k-1}^{po} d\mathbf{x}_{k-1} - \alpha \int_{\mathbb{C}_{p_k}^{\alpha}} p_k^{tr} p_{k-1}^{po} d\mathbf{x}_{k-1}$$
(27)

The first item of (24) is the approximate expectation of the posterior PDF, when the posterior PDF in the posterior CPS at time step k-1 transfers to that of the time step k. A denotes the normalized results of the prior PDF in the $\overline{\mathbb{C}}_{p_k}^{\alpha_{pr}}$. The second item of (24) is another truncation error. The magnitude of ξ_2 is mainly dominated by (27). When $\alpha \to 0$, the two items in (27) tend to 0; then, ξ_2 tends to 0.

The first item of (24) describes the expectation of the continuous posterior PDF in the posterior CPS. After the discretization, we have:

$$\frac{\int_{\mathbb{C}_{p_{k}}^{\alpha}} \mathbf{x}_{k} p_{k}^{li} A d\mathbf{x}_{k}}{(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1})} = \frac{\int_{\mathbb{C}_{p_{k}}^{\alpha}} \mathbf{x}_{k} p_{k}^{li} \sum_{i=1}^{+\infty} \omega_{k-1,i} p_{\mathbf{u}}(\mathbf{x}_{k} - \hat{\mathbf{x}}_{k,i}) d\mathbf{x}_{k}}{(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1})} = \int_{\mathbb{C}_{p_{k}}^{\alpha}} \mathbf{x}_{k} \frac{p_{\mathbf{v}}(\mathbf{y}_{k} - \mathbf{h}(\mathbf{x}_{k})) \sum_{i=1}^{+\infty} \omega_{k-1,i} p_{\mathbf{u}}(\mathbf{x}_{k} - \hat{\mathbf{x}}_{k,i})}{(1-\alpha) p(\mathbf{y}_{k} | \mathbf{y}_{1:k-1})} d\mathbf{x}_{k} = \sum_{j=1}^{+\infty} \omega_{k,j} \mathbf{x}_{k,j} \quad (\mathbf{\tau} \to 0)$$
(28)

Synthesizing (22), (24), and (28):

$$\mathbf{x}_{k,bayes} = \sum_{j=1}^{+\infty} \omega_{k,j} \mathbf{x}_{k,j} + \mathbf{\xi}_1 + \mathbf{\xi}_2$$

Thus,

$$\varepsilon = \mathbf{x}_{k,bayes} - \sum_{i=1}^{N_k} \omega_{k,i} \mathbf{x}_{k,i}$$
$$= \mathbf{\xi}_1 + \mathbf{\xi}_2 + \sum_{j=1}^{+\infty} \omega_{k,j} \mathbf{x}_{k,j} - \sum_{i=1}^{N_k} \omega_{k,i} \mathbf{x}_{k,i}$$

Suppose

$$\boldsymbol{\xi}_3 = \sum_{j=1}^{+\infty} \omega_{k,j} \mathbf{x}_{k,j} - \sum_{i=1}^{N_k} \omega_{k,i} \mathbf{x}_{k,i}$$

Then,

$$\varepsilon = \mathbf{\xi}_1 + \mathbf{\xi}_2 + \mathbf{\xi}_3$$

Both ξ_1 and ξ_2 are truncation errors due to the absence of the low probability space. Specifically, ξ_1 is the state estimation error due to the absence of the $\overline{\mathbb{C}}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}^{\alpha}$. ξ_2 is the state estimation error due to the absence of the $\overline{\mathbb{C}}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$. ξ_3 is the quantization error which is dominated by the sample interval. According to the above derivation, the approximate errors between the CPSF method and Bayesian filter are divided into two parts: the space truncation error (ξ_1 and ξ_2) and quantization error (ξ_3). Note that the magnitude of the space truncation error can be adjusted by the value of α , and the magnitude of the quantization error can be adjusted by the choice of τ . Meanwhile, when $\alpha \to 0$, $\xi_1 \to 0$, $\xi_2 \to 0$, when $\tau \to 0$, $\xi_3 \to 0$. Therefore, the CPSF and Bayesian filter are theoretically consistency. In addition, we have the following Proposition for a particular posterior PDF.

Proposition 2 When the posterior PDF is always a unimodal symmetric function, then $\xi_1 = 0$.

Proof See Appendix C.

Remark 3 The selection of α determines the size of the posterior CPS, which affect the stability of the CPSF method for filtering. The smaller the value of α , the better the stability of CPSF method. Meanwhile, the amount of the computation increases. The magnitude of the sampling interval determines the sample number, which affects the accuracy of the CPSF method for filtering. The smaller the sample interval, the better the accuracy of the CPSF method.

5 Numerical simulation

In this section, the performances including the accuracy, efficiency and robustness of the SI_CPSF algorithm are evaluated using four representative simulation experiments.

5.1 Simulation experiment 1

Consider a classical two-dimensional linear Gaussian filtering system [57]

$$\begin{cases} \mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{u}_{k-1} \\ y_k = \mathbf{H}\mathbf{x}_k + \nu_k \end{cases}$$

where \mathbf{x}_k and y_k denote the state and measurement at time step k, respectively. $\mathbf{F} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$, $\mathbf{H} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, \mathbf{u}_{k-1} and v_k denote the process noise sequence and measurement noise sequence, respectively. In this experiment, the system parameters are set as follows: $\theta = \pi / 18$; $\mathbf{x}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$; $p_{\mathbf{u}} \sim N(0, diag(\begin{bmatrix} 1 & 1 \end{bmatrix}))$; $p_v \sim N(0, 0.1)$. In this sense, the CPS of the process noise is a circle of equal probability. Therefore, the approximate prior CPS is the union of several circles. The likelihood CPS is a space made up of two straight lines. The posterior CPS is the intersection of the prior CPS and the likelihood CPS.

In this linear Gaussian system, the performance of the KF, PF, and the proposed SI_ CPSF was compared. As the optimal Bayesian estimation can be obtained from KF algorithm, the stability of the proposed algorithm is evaluated using the root-mean-square error (RMSE) as follows:



Fig. 4 The sampling mechanism of the SI-CPSF algorithm in one step



Fig. 5 Estimated relative errors compared with KF estimation for the different algorithms

$$\text{RMSE}_{\text{KF}} = \sqrt{\frac{\sum_{i=1}^{monter} \sum_{k=1}^{K} \left(\hat{\mathbf{x}}_{\text{monter},k} - \mathbf{x}_{monter,k}^{KF}\right)^2}{\text{monter} \times K}}$$

For the SI_CPSF algorithm, we choose α =0.0001, and two types of sampling intervals: $\mathbf{\tau} = \begin{bmatrix} 0.8 & 0.8 \end{bmatrix}^T$ and $\mathbf{\tau} = \begin{bmatrix} 0.3 & 0.3 \end{bmatrix}^T$. For the PF (SIR-PF was used in this study), 1000 and 5000 particles are used, respectively. Figure 4 shows the estimated results of the approximate posterior CPS at a certain time step and the process of the weighted grid samples generation and contraction. Figure 5 shows that the SI_CPSF can obtain the result closer to KF comparing with the PF. Figure 6 shows that the SI_CPSF requires a very small number of samples. Figures 6 and 7 show that the required samples number by the SI_CPSF and the computational burden increases when the sampling interval decreases. However, Fig. 7 shows that the time consumption of the SI_CPSF with $\mathbf{\tau} = \begin{bmatrix} 0.3 & 0.3 \end{bmatrix}^T$ is much littler than that of the PF with 5000 particles.



Fig. 6 Samples number for SI_CPSF with different τ



Fig. 7 Computational time cost for the different algorithms

 Table 4
 The performance of different filter algorithms with different parameters in linear/Gaussian system

Algorithms	RMSE _{KF}		Sampling numbers	Time (s)
	<i>x</i> ₁	X ₂		
KF	0	0	_	2.197 × 10 ⁻⁵
PF (particles)	0.4035	0.3959	1000	0.0033
	0.1876	0.1843	5000	0.0931
SI_CPSF				
$\mathbf{\tau} = [0.8, 0.8]^T$	0.0374	0.0066	50	7.358 × 10 ⁻⁴
$\mathbf{\tau} = [0.3, 0.3]^T$	0.0058	0.0056	322	0.0079

To evaluate the stability of the proposed algorithm, we ran 5000 rounds of Monte Carlo. The result can be seen from Table 4. The RMSEs of the KF are the smallest. The RMSEs of the SI_CPSF decrease with the decrease in the sampling interval. The RMSEs of the PF are always larger than that of the SI_CPSF. The SI_CPSF adds the calculation step of the sample weight comparing with the PF. Therefore, the computational burden of the SI_CPSF is bound to be much greater than that of the PF for the same sample number. However, the SI_CPSF just need 50 samples to obtain a better estimate result

than the PF with 5000 particles, when $\mathbf{\tau} = \begin{bmatrix} 0.8 & 0.8 \end{bmatrix}^T$. The large reduction of sample number effectively improves the filtering efficiency. Table 4 shows that the time cost of the SI_CPSF with $\mathbf{\tau} = \begin{bmatrix} 0.8 & 0.8 \end{bmatrix}^T$ is about 1/126 of the PF with 5000 particles. It is worth noting that the posterior PDF always satisfies the Gaussian distribution. Then, we have $\mathbf{\xi}_1=0$ for the SI_CPSF according to Proposition 2. Therefore, the SI_CPSF deviations with Bayesian filter mainly come from the $\mathbf{\xi}_2$ and $\mathbf{\xi}_3$. In addition, Table 4 shows that the RMSEs decrease slightly when the sampling interval decreases from 0.8 to 0.3. It shows that the quantization error has very little effect on RMSEs. At this point, a huge computational cost might be paid for improving the accuracy by reducing the sampling interval. In this case, increasing the posterior CPS by decreasing the α might be worth trying. Of course, a huge computational cost might be paid for improving the filter accuracy when the posterior CPS increases to a certain extent. In short, the improvement of filtering accuracy usually is not linear with the increase in computation burden. However, the SI_CPSF can obviously approximate the recursive Bayesian filtering faster with less computation than the PF in this case.

5.2 Simulation experiment 2

To evaluate the performance of the SI_CPSF in the generical nonlinear system, a typical one-dimensional nonlinear Gaussian system was considered [14]

$$\begin{cases} x_{k+1} = \frac{1}{2}x_k + 25\frac{x_k}{1+x_k^2} + 8\cos(1.2k) + u_k \\ y_k = \frac{x_k^2}{20} + v_k \end{cases}$$

The system parameters were set as follows: $x_0 = 1$; $p_u(x) \sim N(0, 10)$; and $p_v(x) \sim N(0, 1)$. In this sense, the CPS of the process noise is an interval $\left[-x_{\alpha}^u x_{\alpha}^u\right]$, where x_{α}^u is determined by α . The approximate prior CPS is the union of several shifted interval. The likelihood CPS is:

$$\left[-x_{\alpha}^{u}-\sqrt{20y_{k}} \ x_{\alpha}^{u}-\sqrt{20y_{k}} \ \right] \cup \left[-x_{\alpha}^{u}+\sqrt{20y_{k}} \ x_{\alpha}^{u}+\sqrt{20y_{k}} \ \right]$$

The posterior CPS is the intersection of the prior CPS and the likelihood CPS.

The EKF, the PF and the SI_CPSF are used to estimate the sequence of the state $\{x_k\}_{k=1}^{60}$. For the SI_CPSF, we choice $\alpha = [10^{-4} \ 10^{-7}]$, $\tau = [0.3 \ 0.8]$, which forms four sets of parameters in the SI_CPSF. For the PF, 500 and 1000 particles are used respectively. It is worth noting that this state equation is a highly nonlinear function. The like-lihood PDF is bimodal type. Most local inference methods are invalid for this filtering example. Meanwhile, many improved PF (e.g., extended PF, unscented PF) does also not work well [14, 58].

Figure 8 shows that the PF and the SI_CPSF can obtain the estimate results with comparable accuracy. The EKF is much less accurate, which is due to the bimodal character of the likelihood distribution. From the time step 21–23, Fig. 9 shows the change of the PDF in the filtering process. We use the distribution obtained by the PF with 10,000 particles as a reference. It can be seen from Fig. 9 that the posterior CPS can accurately find the high probability region of the posterior PDF. Combine the expression of the posterior PDF in (15), the SI_CPSF can describe the posterior



Fig. 8 Estimated relative errors for different algorithms in one simulation



Fig. 9 Weighted samples at time steps for SI_CPSF



Fig. 10 Computational time cost for the different algorithms

PDF with a small amount of weighted grid samples. Figures 10, 11 and 12 show the performance of filtering based on the Monte Carlo simulation with 30,000 instantiations. Figure 10 shows that the computation of the SI_CPSF is much less than that of the PF. It can be seen from Fig. 11 that the SI_CPSF and the PF have similar state estimation results, which also verifies the consistency of the CPSF method and Bayesian filtering. To verify the robustness of the CPSF method, Fig. 12 (left) shows the cumulative probability function (CDF) of errors of the state results for different algorithms. As the error curves of different algorithms are very close, CDF curve of



Fig. 11 Comparison of the RMSEs for different algorithms



Fig. 12 Comparison of the CDFs between PF and the proposed algorithm

the PF with 1000 particles is used as the reference line, and Fig. 12 (right) shows that the robustness of the PF increases as the particle number increases. (The pink curve is above the green curve in the region with large error (when the error is greater than 16.).) The robustness of the SI_CPSF is always better than that of the two PFs. The robustness of the SI_CPSFs from good to bad is yellow curve, black curve, cyan curve and red curve. This shows that the change of the α has a significant impact on the robustness of filtering. When the error is about 13, the curves of the SI_CDSF from top to bottom are: yellow curve, cyan curve, black curve and red curve. In this region, the sampling interval becomes the dominant factor affecting the error. When the error is about 4, the red curve is the worst, which is influenced by both the sampling interval and the α .

Table 5 shows that the RMSEs of the SI_CPSF are all better than that of the two PFs with lower computational burden. Note that the RMSE of the SI_CPSF with $\alpha = 10^{-4}\tau = 0.3\tau = 0.8$ is equal to the SI_CPSF with $\alpha = 10^{-7}\tau = 0.8$. However,

Algorithms	Parameters	RMSE	Sampling number	Time (s)
EKF	_	12.34	_	1.97 × 10 ⁻⁹
PF	500	4.727	500	8.46×10^{-4}
	1000	4.676	1000	2.47×10^{-3}
SI_CPSF (α / τ)				
10 ⁻⁴	0.8	4.671	20	1.06×10^{-5}
	0.3	4.631	47	1.19 x 10 ⁻⁴
10 ⁻⁷	0.8	4.631	23	7.67 × 10 ^{−5}
	0.3	4.618	61	1.45×10^{-4}
	0.3	4.010	01	1.45 X 10

Table 5 Comparison of the filter accuracy and time cost for different algorithms in one-dimensional nonlinear system

the computational burdens of them are different. This shows that we can further improve the filtering efficiency by adjusting the sampling interval and the α .

5.3 Simulation experiment 3

To evaluate the performance of the SI_CPSF in the multiple dimensions generical nonlinear system, the Lorenz system is considered to verify the performance of the proposed algorithm [59].

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \lambda(-x_1 + x_2) \\ \beta x_1 - x_2 - x_1 x_2 \\ -\gamma x_3 + x_1 x_2 + \end{bmatrix} + \Gamma(t)$$
$$y_k = \sqrt{(x_1(t_k))^2 + (x_2(t_k))^2 + (x_3(t_k))^2} + v_k$$

The system parameters were set as follows: $\mathbf{x}_0 = [-100, 100 \ 50]^T$, $\lambda = 10, \beta = 28, \gamma = 8/3, p_\mathbf{u} \sim N(0, diag([1 1 1 1])); p_\nu \sim N(0, 1)$. In this sense, the CPS of the process noise is a sphere of equal probability. Therefore, the approximate prior CPS is the union of several spheres. The likelihood CPS is a ring of two concentric spheres: $y_k - r_\alpha \leq r \leq y_k + r_\alpha$, where r_α is determined by the α . The PF and the SI_CPSF are used to estimate the sequence of the state $\{\mathbf{x}_k\}_{k=1}^{100}$. For the PF, the particle numbers are 500 and 10,000, respectively. We choice $\alpha = 10^{-4}, \tau = [2.5 \ 2.5 \ 2.5 \]^T$ and $\tau = [1.5 \ 1.5 \ 1.5 \]^T$, which form two sets of parameters for the SI_CPSF. This experiment is also a typical general nonlinear filtering scenario. The filtering effect of most local inference and improved PFs is not good.

Figure 13 shows the weighted grid samples, generated in the posterior CPS, for the SI_CPSF with different sampling intervals. As we can see that the samples are evenly distributed in the space, their probability information is completely focused on the weight. Figure 14 shows that the RMSEs of the SI_CPSF with two sets of sample interval are similar to that of the PF with 500, 10,000 particles, respectively. Figure 15 shows that the time consumption of the SI_CPSF with $\tau = 1.5$ is less than that of the PF with 10,000 particles. Figure 16 shows the number of samples required for the SI_CPSF at different sampling intervals. It shows that the number of samples required by the SI_CPSF under these two sets of sample interval is much less than that of the PF. This is an important



Fig. 13 The sampling mechanism of the SI_CPSF algorithm in one step



Fig. 14 Comparison of the RMSEs for different algorithms



Fig. 15 Computational time cost for the different algorithms

reason that the SI_CPSF is superior to the PF in filtering efficiency, which can be seen from Fig. 15. Similar to the experiment 2, we obtain Fig. 18 according to Fig. 17. The robustness of the two SI_CPSFs is better than that of the two PFs, respectively.

Table 6 shows the accuracy and effectiveness of the filtering based on the Monte Carlo simulation with 5000 instantiations. It shows that the SI_CPSF with $\tau = 1.5$ can obtain a better accuracy of filtering using a lower computation burden compared



Fig. 16 The average samples number for SI_CPSF with different parameters



Fig. 17 Comparison of the CDFs for PF and the proposed algorithm



Fig. 18 Comparison of the CDFs between PF and the proposed algorithm

with the PF with 10,000 particles. Compared with the PF, the simulation shows that the SI_CPSF has a faster convergence rate in the process of approaching the Bayesian filtering.

Algorithms	RMSE	Sampling number	Time (s)	
PF	5.07	500	0.0045	
	4.95	10,000	0.2725	
SI_CPSF $\tau = [2.5 \ 2.5 \ 2.5]^T$	5.04	158	0.0058	
SI_CPSF $\tau = [1.5 \ 1.5 \ 1.5]^T$	4.94	725	0.0814	

Table 6 Comparison of the filter accuracy and time cost for different algorithms in threedimensional nonlinear system



Fig. 19 The average samples number for SI_CPSF with different parameters for target tracking of the UAV with radar

5.4 Simulation experiment 4

To evaluate the performance of the SI_CPSF in a real-life scenario, the proposed algorithm is verified by the target tracking of the UAV with radar. The dynamic models adopted in this work is:

$$\mathbf{p}_{k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{k-1} + \begin{bmatrix} v_{k-1}^{x} \Delta t \\ v_{k-1}^{y} \Delta t \\ v_{k-1}^{z} \Delta t \end{bmatrix} + \mathbf{u}_{k-1}$$

where $\mathbf{p}_k = [p_k^x, p_k^y, p_k^z]^T$, p_k^x , p_k^y and p_k^z are the position of the aircraft in three directions at time step k; v_k^x , v_k^y and v_k^z are real velocities corresponding to these two position directions (provided by the IMU) at time step k. \mathbf{u}_{k-1} is zero-mean Gaussian processes characterizing the IMU measurement noise.

The dynamic measurement model characterizing the radar system is:

$$Z_{k} = \sqrt{\left(p_{k}^{x} - p_{radar}^{x}\right)^{2} + \left(p_{k}^{y} - p_{radar}^{y}\right)^{2} + \left(p_{k}^{z} - p_{radar}^{z}\right)^{2}} + \nu_{k}$$

where $[p_{radar}^{x}, p_{radar}^{y}, p_{radar}^{z}]^{T}$ is the coordinates of the radar and is known. v_{k} is a zeromean Gaussian processes characterizing the radar measurement noise.

The system parameters were set as follows: $\mathbf{p}_0 = [2600, 2800, 800]^T$, $\mathbf{p}_{radar} = [3500, 2300, 0]^T$, $\mathbf{u}_{k-1} \sim N(0, diag([1 \ 1 \ 1 \ 1])); \nu_k \sim N(0, 0.1)$. We choice



Fig. 20 Comparison of the CDFs for PF and the proposed algorithm for target tracking of the UAV with radar

 Table 7
 Comparison of the filter accuracy and time cost for different algorithms in target tracking scenario

Algorithms	RMSE	Sampling number	Time (s)
PF	3.5698	3000	0.0251
	3.4595	10,000	0.2386
SI_CPSF $\boldsymbol{\tau} = [1 \ 1 \ 1]^T$	3.4285	1549	0.0347
SI_CPSF $\tau = [0.8 \ 0.8 \ 0.8]^T$	3.2829	3156	0.1876

 $\alpha = 10^{-4}$, $\tau = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ and $\tau = \begin{bmatrix} 0.8 & 0.8 & 0.8 \end{bmatrix}^T$, which form two sets of parameters for the SI CPSF. For the PF, the particle numbers are 500 and 10,000, respectively. In this scenario, we ran 100 rounds of Monte Carlo simulations. The average number of particles used by the proposed algorithm during the recursive process is shown in Fig. 19. It can be seen that, in the initial stages, the number of particles steadily increases and eventually stabilizes. This is due to the continuous accumulation of IMU errors during the target tracking process, while observations continually correct its state. Therefore, the variance of the posterior distribution of the target state is a process of continuous amplification that eventually reaches stability. The proposed algorithm closely aligns with this variation process. Figure 20 shows the RMSEs of CPSFs and PFs based on the Monte Carlo simulation with 1000 instantiations. It can be observed that increasing the number of particles in PF from 3000 to 10,000 has a negligible impact on the filtering accuracy. However, the computational cost experiences a significant increase. When the CPSF sampling interval is set to 1, its performance is similar to that of the PF with 10 K particle number; when the CPSF sampling interval is set to 0.8, its performance is significantly better than that of the PF with 10 K particle number. For further comparison, we provide Table 7. It is evident that when the sampling interval is set to 1, CPSF passes an average of just over 1500 samples, and its computational cost is much lower than that of PF with 10 K particles. When the sampling interval is set to 0.8, CPSF exhibits an improvement of nearly 10% in filtering accuracy compared to PF with 10 K particles, and its computational cost is lower than that of PF.

6 Conclusion

The CPSF method was proposed to solve the generic nonlinear filtering problem in this paper. This method has absorbed the idea of the SMC method, where the high probability region of the posterior PDF dominates the performance of filtering. Meanwhile, the transfer of the posterior PDF was based on the idea of the numerical integration in GF and the local inference. The posterior CPS was used to select the high probability region. The weighted grid samples in the posterior CPS were calculated by the PST. The approximate expression of the posterior PDF, as key for the PST implementation, was obtained by using the properties of the impulse function in the integral operation. On this basis, we proposed the CPSF framework for solving the generic nonlinear filtering problem. As finding the posterior CPS is difficult directly, SI method was proposed to obtain the approximate posterior CPS, which formed the SI_CPSF algorithm. The approximate error from the CPSF framework to the recursive Bayesian filter is analyzed. The consistency of the CPSF framework to the recursive Bayesian filter is proved in this paper. Four typical simulation experiments were used to evaluate the performance of the proposed method.

Appendix A

Proof of Theorem 1

Proof As $\tau \to 0$, the size of each block $\Delta \to 0$, where $\Delta = \tau_1 \times \tau_2 \times \cdots \times \tau_D$. When $\alpha \to 0$, $\mathbb{C}^{\alpha}_{p(\mathbf{x})} \to \mathbb{C}^{0}_{p(\mathbf{x})}$, and we have:

$$\lim_{\alpha, \mathbf{\tau} \to \mathbf{0}} \sum_{l=1}^{+\infty} p(\mathbf{x}_l) \Delta = \int p(\mathbf{x}) d\mathbf{x} = 1$$

according to the property of the PDF. $p(\mathbf{x})$ can be expressed as follows:

$$p(\mathbf{x}) = \frac{p(\mathbf{x})}{\lim_{\alpha, \tau \to \mathbf{0}} \sum_{l=1}^{+\infty} p(\hat{\mathbf{x}}_l) \Delta}$$

Substitute (8) into the above equation and we have:

$$p(\mathbf{x}) = \frac{1}{\Delta} \lim_{\alpha, \tau \to \mathbf{0}} \sum_{l=1}^{+\infty} \omega_l \delta \left(\mathbf{x} - \hat{\mathbf{x}}_l \right)$$

As Δ is a constant,

$$p(\mathbf{x}) \propto \lim_{\alpha, \tau \to \mathbf{0}} \sum_{l=1}^{+\infty} \omega_l \delta(\mathbf{x} - \hat{\mathbf{x}}_l)$$

Appendix B

Proof of Proposition 1

Proof Each sample $\hat{\mathbf{x}}_{k-1,n}$ in the $\hat{\mathbf{X}}_{k-1}$ represents a subspace of the $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$. The $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ is the union of each subspace. For an arbitrary sample $\hat{\mathbf{x}}_{k-1,n}$, it will be transferred to $\mathbf{x}_{k,n}$ without the process noise, where $\mathbf{x}_{k,n} = \mathbf{f}(\hat{\mathbf{x}}_{k-1,n})$. After introducing the additive noise \mathbf{u}_k , $\hat{\mathbf{x}}_{k-1,n}$ will be transferred to the $\hat{\mathbf{x}}_{k-1,n} + \mathbb{C}_{p_n}^{\alpha}$ under the confidence

 $1 - \alpha$. When $\tau \to \mathbf{0}$, the subspace, represented by $\hat{\mathbf{x}}_{k-1,n}$, approximates to be transferred to $\hat{\mathbf{x}}_{k-1,n} + \mathbb{C}_{p_{\mathbf{u}}}^{\alpha}$ under the confidence $1 - \alpha$.

According to the addition property, the $\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}$ will be transferred to the $\tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{\alpha}$ under the confidence $1 - \alpha$, where:

$$\tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{\alpha} = \bigcup_{n=1}^{N_{k-1}} \left(\vec{\mathbf{x}}_{k,n} + \mathbb{C}_{p_{\mathbf{u}}}^{\alpha} \right)$$

Consider:

$$\int_{\mathbb{C}_{p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})}^{\alpha}} p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} = 1 - \alpha$$

according to Definition 1. Then,

$$P\left(\mathbf{x}_{k} \in \mathbb{C}_{p\left(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}\right)}^{\alpha}\right) = 1 - \alpha$$

and

$$P\left(\mathbf{x}_{k} \in \tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{\alpha}\right) \geq (1-\alpha)^{2}$$

For the likelihood CPS $\mathbb{C}_{p(\mathbf{y}_k|\mathbf{x}_k)}^{\alpha}$, we have:

$$\int_{\mathbb{C}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}^{\alpha}} p(\mathbf{y}_{k}|\mathbf{x}_{k}) d\mathbf{x}_{k} = 1 - \alpha$$

which means:

$$P\left(\mathbf{x}_{k} \in \mathbb{C}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}^{\alpha}\right) = 1 - \alpha$$

The complete prior state space, likelihood state space and posterior state space belong to the same space. That is:

$$\mathbb{C}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{0} = \mathbb{C}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{0} = \mathbb{C}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{0}$$

As:

$$\tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})} = \tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})} \cap \mathbb{C}^{\alpha}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}$$

We rewrite it as:

$$\tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})} = \mathbb{Q} + \left(\mathbb{C}^{0}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})} - \tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}\right) \cap \left(\mathbb{C}^{0}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})} - \mathbb{C}^{\alpha}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}\right)$$

where

$$\mathbb{Q} = \tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_k|\mathbf{y}_{1:k-1})} + \mathbb{C}^{\alpha}_{p(\mathbf{y}_k|\mathbf{x}_k)} - \mathbb{C}^{0}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}$$

Then,

$$P(\mathbf{x}_{k} \in \mathbb{Q})$$

$$= P\left(\mathbf{x}_{k} \in \tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k-1})}^{\alpha}\right) + P\left(\mathbf{x}_{k} \in \mathbb{C}_{p(\mathbf{y}_{k}|\mathbf{x}_{k})}^{\alpha}\right) + P\left(\mathbf{x}_{k} \in \mathbb{C}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{0}\right)$$

$$\geq (1-\alpha)^{2} + (1-\alpha) - 1$$

$$= 1 - \left(3\alpha - \alpha^{2}\right)$$

As

$$\mathbb{Q} \subseteq \tilde{\mathbb{C}}^{\alpha}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}$$

Meanwhile, we can obtain the optimal state estimation from the expectation of the $p(\mathbf{x}_k | \mathbf{y}_{1:k})$. Then,

$$P\left(\mathbf{x}_{k} \in \tilde{\mathbb{C}}_{p(\mathbf{x}_{k}|\mathbf{y}_{1:k})}^{\alpha}\right) \geq P(\mathbf{x}_{k} \in \mathbb{Q}) \geq 1 - \left(3\alpha - \alpha^{2}\right)$$

Therefore, the lower bound of the confidence of $\tilde{\mathbb{C}}_{p(\mathbf{x}_k|\mathbf{y}_{1:k})}^{\alpha}$ approximates to $1 - (3\alpha - \alpha^2)$.

Appendix C

Proof of Proposition 2 *Proof* As:

$$\int_{\overline{\mathbb{C}}_{p_{k}^{o}}} p_{k}^{po} d\mathbf{x}_{k} = \alpha$$

according to Definition 1. When the posterior PDF is always a unimodal symmetric function, then

$$\frac{\int_{\overline{\mathbb{C}}_{p_k^{opo}}}^{\alpha} \mathbf{x}_k p_k^{po} d\mathbf{x}_k}{\int_{\overline{\mathbb{C}}_{p_k^{opo}}}^{\alpha} p_k^{po} d\mathbf{x}_k} = \frac{\int_{\mathbb{C}_{p_o}}^{\alpha} \mathbf{x}_k p_k^{po} d\mathbf{x}_k}{\int_{\mathbb{C}_{p_k^{opo}}}^{\alpha} p_k^{po} d\mathbf{x}_k} = \mathbf{x}_{k, bayes}$$

Furthermore,

$$\int_{\overline{\mathbb{C}}_{p_k^{p_o}}^{\alpha_{p_o}}} \mathbf{x}_k p_k^{p_o} d\mathbf{x}_k = \alpha \mathbf{x}_{k, bayes}$$
$$\int_{\mathbb{C}_{p_k^{p_o}}^{\alpha_{p_o}}} \mathbf{x}_k p_k^{p_o} d\mathbf{x}_k = (1-\alpha) \mathbf{x}_{k, bayes}$$

Then,

$$\begin{aligned} \mathbf{\xi}_1 &= \alpha \mathbf{x}_{k, bayes} - \frac{\alpha}{1-\alpha} (1-\alpha) \mathbf{x}_{k, bayes} \\ &= 0 \end{aligned}$$

Authors contributions

XQ contributed to conceptualization, data curation, methodology, validation, and writing—original draft, and provided software. RX was involved in conceptualization, investigation, methodology, funding acquisition, and writing—review and editing. YZ contributed to funding acquisition and supervision.

Funding

This work was supported by the Doctoral research start-up fund S020318015/028.

Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication All authors agree to publish the submitted paper in this journal.

Competing interests

The authors declare that they have no competing interests.

Received: 16 February 2023 Accepted: 8 February 2024

Published online: 19 February 2024

References

- 1. H. Kushner, Approximations to optimal nonlinear filters. IEEE Trans. Autom. Control 12(5), 546–556 (1967)
- 2. B.D. Anderson, J.B. Moore, Optimal filtering (Prentice-Hall, Englewood Cliffs, 1979)
- S. Gultekin, J. Paisley, Nonlinear Kalman filtering with divergence minimization. IEEE Trans. Signal Process. 65(23), 6319–6331 (2017)
- S. Schlupkothen, G. Ascheid, Multiple particle filtering for tracking wireless agents via Monte Carlo likelihood approximation. EURASIP J. Adv. Signal Process. 2019(52), 1–20 (2019)
- W.R. Gilks, C. Berzuini, Following a moving target-Monte Carlo inference for dynamic Bayesian models. J. R. Statist. Soc. B 63(1), 127–146 (2001)
- R.E. Kalman, A new approach to linear filtering and prediction problems. Trans. ASME Ser. D J. Basic Eng. 82, 35–45 (1960)
- R. Dehghannasiri, M.S. Esfahani, E.R. Dougherty, Intrinsically Bayesian robust Kalman filter: an innovation process approach. IEEE Trans. Signal Process. 65(10), 2531–2546 (2017)
- R. Dehghannasiri, M.S. Esfahani, X. Qian, E.R. Dougherty, Optimal Bayesian Kalman filtering with prior update. IEEE Trans. Signal Process. 66(8), 1982–1996 (2018)
- S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50(2), 174–188 (2002)
- D.S. Kalogerias, A.P. Petropulu, Grid based nonlinear filtering revisited: Recursive estimation & asymptotic optimality. IEEE Trans. Signal Process. 64(16), 4244–4259 (Aug.2016)
- T. Schon, F. Gustafsson, P. Nordlund, Marginalized particle filters for mixed linear/nonlinear state-space models. IEEE Trans. Signal Process. 53(7), 2279–2289 (2005)
- 12. K.J. Kim, M.-O. Pun, R.A. Iltis, Joint carrier frequency offset and channel estimation for uplink MIMO-OFDMA systems using parallel Schmidt Rao-Blackwellized particle filters. IEEE Trans. Commun. **58**(9), 2697–2708 (2010)
- I. Arasaratnam, S. Haykin, R.J. Elliot, Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. Proc. IEEE 95(5), 953–977 (May2007)
- 14. S. Mazuelas, Y. Shen, M.Z. Win, Belief condensation filtering. IEEE Trans. Signal Process. 61(18), 4403–4415 (2013)
- 15. S. Julier, J. Uhlmann, A new extension of the Kalman filter to nonlinear systems. Proc. SPIE **3068**, 182–193 (1997)
- 16. B. Ristic, M.S. Arulampalam, N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications* (Artech House, Norwood, 2004)
- 17. Y.B. Shalom, X.-R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation (Wiley, New York, 2001)
- 18. S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation. Proc. IEEE 92(3), 401–422 (2004)
- S.J. Julier, J.K. Ulhmann, H.F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans. Autom. Control 45(3), 472–482 (2000)
- 20. I. Arasaratnam, S. Haykin, Cubature Kalman filters. IEEE Trans. Autom. Control 54(6), 1254–1269 (2009)
- I. Arasaratnam, S. Haykin, T.R. Hurd, Cubature Kalman filtering for continuous-discrete systems: theory and simulations. IEEE Trans. Signal Process. 58(10), 4977–4993 (Oct.2010)
- G. Revach, N. Shlezinger, X. Ni, A.L. Escoriza, R.J. Van Sloun, Y.C. Eldar, KalmanNet: neural network aided Kalman filtering for partially known dynamics. IEEE Trans. Signal Process. 70, 1532–1547 (2022)
- K. Radnosrati, G. Hendeby, F. Gustafsson, Exploring Positive Noise in Estimation Theory. IEEE Trans. Signal Process. 68, 3590–3602 (2020)

- 24. S. Mazuelas, Y. Shen, A. Pérez, Generalized maximum entropy for supervised classification. IEEE Trans. Information Theory **68**(4), 2530–2550 (2022)
- 25. A. Doucet, J. de Freitas, N. Gordon, Sequential Monte Carlo in practice (Cambridge University Press, Cambridge, 2001)
- 26. A. Doucet, J.F.G. de Freitas, N.J. Gordon, An introduction to sequential Monte Carlo methods, in *Sequential Monte Carlo Methods in Practice* (Springer, New York, 2001)
- C. Andrieu, A. Doucet, Particle filtering for partially observed Gaussian state space models. J. Roy. Stat. Soc. Ser. B 64(4), 827–836 (2002)
- F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, P.J. Nordlund, Particle filters for positioning, navigation and tracking. IEEE Trans. Signal Process. 50(2), 425–437 (2002)
- N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. EEE Proc. F Radar Signal Process. 140(2), 107–113 (1993)
- O. Hlinka, F. Hlawatsch, P.M. Djuric, Distributed particle filtering in agent networks: a survey, classification, and comparison. IEEE Signal Process. Mag. 30(1), 61–81 (2013)
- A. Mohammadi, A. Asif, Distributed particle filter implementation with intermittent/irregular consensus convergence. IEEE Trans. Signal Process. 61(10), 2572–2587 (2013)
- 32. F. Daum, J. Huang, Curse of dimensionality and particle filters. Proc. IEEE Aerosp. Conf. 4, 1979–1993 (2003)
- T. Bengtsson, P. Bickel, B. Li, Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems. Probab. Stat. Essays Honor David A Freedman 2, 316–334 (2008)
- K. Chen, S. Werner, A. Kuh, Y. Huang, Nonlinear adaptive filtering with kernel set-membership approach. IEEE Trans. Signal Process. 68, 1515–1528 (2020). https://doi.org/10.1109/TSP.2020.2975370
- T. Shnitzer, R. Talmon, J. Slotine, Diffusion maps Kalman filter for a class of systems with gradient flows. IEEE Trans. Signal Process. 68, 2739–2753 (2020)
- X. Qiang, Y. Zhu, R. Xue, SVRPF: an improved particle filter for a nonlinear/non-Gaussian environment. IEEE Access 7, 151638–151651 (2019)
- D. Berberidis, G.B. Giannakis, Data sketching for large-scale Kalman filtering. IEEE Trans. Signal Process. 65(14), 3688–3701 (2017)
- H. Yazdanpanah, M.V. Lima, P.S. Diniz, On the robustness of set-membership adaptive filtering algorithms. EURASIP J. Adv. Signal Process. 2017(1), 1–12 (2017)
- T. Li, T.P. Sattar, S. Sun, Deterministic resampling: unbiased sampling to avoid sample impoverishment in particle filters. Signal Process. 92(7), 1637–1645 (2012)
- M. Bolic, P.M. Djuric, S. Hong, Resampling algorithms and frameworks for distributed particle filters. IEEE Trans. Signal Process. 53(7), 2442–2450 (2005)
- 41. M. Bolic, P.M. Djuric, S. Hong, Resampling algorithms for particle filters: a computational complexity perspective. EURASIP J. Appl. Signal Process. **2004**(15), 2267–2277 (2004)
- T. Li, M. Bolic, P.M. Djuric, Resampling methods for particle filtering: classification implementation and strategies. IEEE Signal Process. Mag. 32(3), 70–86 (2015)
- R. Douc, O. Capp, Comparison of resampling schemes for particle filtering, in Proceedings of the 4th International Symposium Image and Signal Processing and Analysis, pp. 64–69 (2005).
- A.A. Nasir, S. Durrani, R.A. Kennedy, Particle filters for joint timing and carrier estimation: Improved resampling guidelines and weighted Bayesian Cramer-Rao bounds. IEEE Trans. Commun. 60(5), 1407–1419 (2012)
- R. Rabiee, X. Zhong, Y. Yan, W.P. Tay, LaIF: A lane-level self-positioning scheme for vehicles in GNSS-denied environments. IEEE Trans. Intell. Transp. Syst. 20(8), 2944–2961 (2019)
- R. Merwe, A. Doucet, N. Freitas, E. Wan, The unscented particle filter, Technical Report CUED/F-INFENG/TR 380, Engineering Department, University of Cambridge (2000).
- L. Wei, Y. Wang, P. Chen, A particle filter-based approach for vehicle trajectory reconstruction using sparse probe data. IEEE Trans. Intell. Transp. Syst. 22(5), 2878–2890 (2021)
- S.K. Pang, J.D.B. Nelson, S.J. Godsill, N. Kingsbury, Video tracking using dual-tree wavelet polar matching and Rao-Blackwellised particle filter. EURASIP J. Adv. Signal Process. 2009(1), 1–13 (2009)
- Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, V. Gupta, State estimation in electric power grids. IEEE Signal Process. Mag. 29(5), 33–43 (Sep.2012)
- Y. Choe, J.W. Song, C.G. Park, Lightweight marginalized particle filtering with enhanced consistency for terrain referenced navigation. IEEE Trans. Aerosp. Electron. Syst. 58(3), 2493–2504 (2022)
- 51. L. Martino, V. Elvira, Compressed Monte Carlo with application in particle filtering. Inf. Sci. 553, 331–352 (2021)
- 52. H. J. Kushner, Numerical approximations to optimal nonlinear filters, in D. Crisan and B. Rozovski, Eds. *The Oxford Handbook of Nonlinear Filtering*, Oxford, ch. 28 (2011).
- G. Pagès, H. Pham, Optimal quantization methods for nonlinear filtering with discrete-time observations. Bernoulli 11(5), 893–932 (2005)
- U.A. Khan, J.M.F. Moura, Distributing the Kalman filter for large-scale systems. IEEE Trans. Signal Process. 56(10), 4919–4935 (2008)
- M.S. Mahmud, J.Z. Huang, S. Salloum, T.Z. Emara, K. Sadatdiynov, A survey of data partitioning and sampling methods to support big data analysis. Big Data Min. Anal. 3(2), 85–101 (2020)
- S. Salloum, J.Z. Huang, Y. He, Random Sample partition: a distributed data model for big data analysis. IEEE Trans. Ind. Inform. 15(11), 5846–5854 (2019)
- 57. B. Chen, X. Liu, H. Zhao, J.C. Prncipe, Maximum correntropy Kalman filter. Automatica 76, 70–77 (2017)
- 58. Z. Chen, Bayesian filtering: from Kalman filters to particle filters, and beyond. Statistics 182(1), 1–69 (2003)
- G. Terejanu, P. Singla, T. Singh, P.D. Scott, Adaptive Gaussian sum filter for nonlinear Bayesian estimation. IEEE Trans. Autom. Control 56(9), 2151–2156 (2011)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.