An Improved Gaussian Particle Filter Algorithm Using KLD-Sampling

*ZHOU Zhaihe**, *ZHONG Yulu*, *ZENG Qingxi*, *TIAN Xiangrui*

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China

(Received 26 June 2019; revised 27 December 2019; accepted 25 July 2020)

Abstract: To adjust the samples of filtering adaptively, an improved Gaussian particle filter algorithm based on Kullback-Leibler divergence (KLD)-sampling (KLGPF) is proposed in this paper. During the process of sampling, the algorithm calculates the KLD to adjust the size of the particle set between the discrete probability density function of particles and the true posterior probability density function. KLGPF has significant effect when the noise obeys Gaussian distribution and the statistical characteristics of noise change abruptly. Simulation results show that KLGPF could maintain a good estimation effect when the noise statistics changes abruptly. Compared with the particle filter algorithm using KLD-sampling (KLPF), the speed of KLGPF increases by 28% under the same conditions.

Key words: particle filter; Gaussian particle filter; KLD-sampling; noise mutation; adaptive particle numbers

CLC number: TP274 **Document code:** A **Article ID:** 1005-1120(2020)04-0607-08

0 Introduction

Nonlinear filtering problems may occur in many fields, including target tracking[1], strapdown inertial navigation system (SINS), and attitude estimation[2]. The particle filter (PF) algorithm proposed by Doucet is a filtering algorithm based on the Monte Carlo method, whose dimension, confidence, sampling and many other issues have received extensive attention and research[3-4]. The resampling strategy in PF algorithm will directly affect the performance of filtering. Based on the framework of PF, Gaussian particle filter (GPF)[5] uses the Gaussian distribution to approximate the prior and posterior distribution of the state, which is a kind of resampling-free filtering algorithm and has better real-time performance than PF[6].

In fact, the noise of the system is not constant all the time[7]. For example, in attitude estimation, the noise of gyroscope is affected by air pressure and temperature. Similarly, in speech recognition[8], it is affected by the surrounding environment. In such cases, GPF and PF frequently have divergence due to the sudden change of noise. Over the past years, adaptive particle filters which use the Kullback-Leibler divergence (KLD)-sampling have been applied in many areas[9-11]. Hereinafter, the method is called KLPF. However, when it is applied to other occasions, due to the resampling of PF, the real-time performance of KLPF is poor.

Aiming at solving the above problem, an improved Gaussian particle filter algorithm named KLGPF is proposed using KLD-sampling in this paper. By introducing the KLD-sampling strategy, the algorithm calculates the KLD between the discrete probability density function and the true posterior probability density function, which is represented by the particle to adapt to the number of samples[12]. In contrast to KLPF, the proposed algorithm is integrated into GPF so that the real-time performance has great improvement.

The remainder of this paper is organized as fol-

*Corresponding author, E-mail address: zzh_nuaa@126.com.

How to cite this article: ZHOU Zhaihe, ZHONG Yulu, ZENG Qingxi, et al. An improved Gaussian particle filter algorithm using KLD-sampling[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(4): 607-614.

http://dx.doi.org/10.16356/j.1005-1120.2020.04.011

lows. In the next section, the background information about KLGPF is reviewed. In Section 2, the improved Gaussian particle filtering using KLD-sampling is introduced. Simulation results are presented in Section 3 and the conclusions are drawn in Section 4.

1 Background

1.1 Gaussian particle filter

For the abovementioned nonlinear filtering problem, most dynamic state space (DSS) model with such problems can be written as[13]

$$\begin{cases} \boldsymbol{x}_n = f(\boldsymbol{x}_{n-1}, \boldsymbol{u}_n) & \text{Process equation} \\ \boldsymbol{y}_n = h(\boldsymbol{x}_n, \boldsymbol{v}_n) & \text{Observation equation} \end{cases} \tag{1}$$

where $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ are state and observational variables, respectively; and $\boldsymbol{u}_n$ and $\boldsymbol{v}_n$ white noises. The nonlinear functions of the system are represented by $f(\cdot)$ and $h(\cdot)$. n is a timestamp depicting system time.

The PF method is to get M particles from the importance probability density $\pi(\cdot)$. Samples $\{\boldsymbol{x}_n^i\}_{i=1}^M$ can be used to describe the importance probability density function $\pi(\cdot)$ of the state $\boldsymbol{x}_n$ at n time[14].

From $w^{(i)} = (p(\boldsymbol{x}_n^i)/\pi(\boldsymbol{x}_n^i))$, the weighted value of the particles can be obtained, where $p(\boldsymbol{x}_n)$ is a posterior distribution. The posterior distribution can be represented by a sample set $\{\boldsymbol{x}, W\}$.

The estimate of state $\boldsymbol{x}_n$

$$E_p(\boldsymbol{x}_n) = \int \boldsymbol{x}_n \, p(\boldsymbol{x}_n) \mathrm{d}\boldsymbol{x}_n \tag{2}$$

can be calculated as

$$\hat{E}_p(\boldsymbol{x}_n) = \frac{\sum_{i=1}^{M} w^{(i)} \boldsymbol{x}_n^{(i)}}{\sum_{i=1}^{M} w^{(i)}} \tag{3}$$

It is obtained from the strong law of large numbers that $\hat{E}_p(\boldsymbol{x}_n) \to E_p(\boldsymbol{x}_n)$ with $M \to \infty$. Then the approximation of the posterior probability density can be written as

$$p(\boldsymbol{x}_n) \approx w_n^i \delta(\boldsymbol{x}_n - \boldsymbol{x}_n^i) \tag{4}$$

where $\delta(\cdot)$ is a Dirac delta function.

Assume that the distribution of state $\boldsymbol{x}_n$ at initial time is $p(\boldsymbol{x}_1 \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{x}_1; \bar{\boldsymbol{\mu}}_1, \overline{\boldsymbol{\Sigma}}_1)$, where $\mathcal{N}(\cdot)$ is a Gaussian distribution that can be expressed as

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-m/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right) \tag{5}$$

where $\bar{\boldsymbol{\mu}}_1$ and $\overline{\boldsymbol{\Sigma}}_1$ are decided by prior information.

In general, GPF is divided into two processes including measurement update and time update, which will be introduced in the following section.

1.1.1 Measurement update

Each particle of set $\{\boldsymbol{x}_n^{(i)}\}_{i=1}^M$ is given a weight obtained from the importance probability density $\pi(\boldsymbol{x}_n \mid \boldsymbol{y}_{0:n})$.

$$\overline{w}_n^{(i)} = \frac{p(\boldsymbol{y}_n \mid \boldsymbol{x}_n^{(i)}) \mathcal{N}(\boldsymbol{x}_n = \boldsymbol{x}_n^{(i)}; \overline{\boldsymbol{\mu}}_n, \overline{\boldsymbol{\Sigma}}_n)}{\pi(\boldsymbol{x}_n^{(i)} \mid \boldsymbol{y}_{0:n})} \tag{6}$$

Then, the weights are normalized as Eq.(7) to ensure the correctness of the weighted sum.

$$w_n^{(i)} = \overline{w}_n^{(i)} \Big/ \sum_{i=1}^{M} \overline{w}_n^{(i)} \tag{7}$$

Finally, the posterior distribution of the state $\boldsymbol{x}_n$ is approximated to the Gaussian distribution, and the mean and variance of the Gaussian distribution are calculated as

$$\begin{cases} \boldsymbol{\mu}_n = \sum_{i=1}^{M} w_n^{(i)} \boldsymbol{x}_n^{(i)} \\ \boldsymbol{\Sigma}_n = \sum_{i=1}^{M} w_n^{(i)} (\boldsymbol{\mu}_n - \boldsymbol{x}_n^{(i)})(\boldsymbol{\mu}_n - \boldsymbol{x}_n^{(i)})^{\mathrm{H}} \end{cases} \tag{8}$$

1.1.2 Time update

Update the state of each particle of set $\{\boldsymbol{x}_n^{(i)}\}_{i=1}^M$ to get the updated particle, which is obtained from the posterior probability distribution $\mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. The predicted probability density function is approximated to the Gaussian distribution, and the mean and variance of the Gaussian distribution are calculated as

$$\begin{cases} \boldsymbol{\mu}_n = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_n^{(i)} \\ \boldsymbol{\Sigma}_n = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{\mu}_n - \boldsymbol{x}_n^{(i)})(\boldsymbol{\mu}_n - \boldsymbol{x}_n^{(i)})^{\mathrm{H}} \end{cases} \tag{9}$$

1.2 KLD-sampling

The KLD-sampling method keeps the KLD between two probability density functions under a

threshold, where KLD is defined as

$$D_{\mathrm{KL}}(P /\!/ Q)=\sum_{i=1}^{k}P(i)\log\frac{P(i)}{Q(i)} \tag{10}$$

Therefore, the posterior distribution is first viewed as a discrete piecewise function. From this discrete distribution, M samples $\underset{\sim}{x}=(x_1,x_2,\cdots,x_k)$ are obtained, which falls into a different interval.

Obviously, $\underset{\sim}{x}$ conforms to multinomial distribution that is denoted as $\underset{\sim}{x}\sim PN(M,\underset{\sim}{p})$, where $\underset{\sim}{p}=p_1,p_2,\cdots,p_k$ is the corresponding probability for each bins. The maximum likelihood estimate of p_i is $\hat{p}_i=x_i/M$, $i=1,2,\cdots,k$.

Substitute the maximum likelihood estimation probability and the real posterior distribution into the KLD equation as

$$D_{\mathrm{KL}}(\underset{\sim}{\hat{p}} /\!/ \underset{\sim}{\hat{p}})=\sum_{i=1}^{k}\hat{p}(i)\log\frac{\hat{p}(i)}{p(i)} \tag{11}$$

For $M\to\infty$, Eq.(11) can be written as

$$2MD_{\mathrm{KL}}(\underset{\sim}{\hat{p}} /\!/ \underset{\sim}{\hat{p}})\to\chi^2_{k-1} \tag{12}$$

where χ^2_{k-1} is a chi-square distribution with $k-1$ degrees of freedom.

After performing interval estimation, the particle size M can be computed by

$$M=\frac{1}{2e}\chi^2_{k-1,1-\delta} \tag{13}$$

A good approximation was provided by the Wilson-Hilferty transformation[15], which yields

$$M=\frac{1}{2e}\chi^2_{k-1,1-\delta}\approx\frac{k-1}{2e}\left\{1-\frac{2}{9(k-1)}+\sqrt{\frac{2}{9(k-1)}}z_{1-\delta}\right\}^3 \tag{14}$$

where $z_{1-\delta}$ is the upper $1-\delta$ quantile of the standard normal distribution and e the threshold of the KLD.

2 Gaussian Particle Filter Using KLD-Sampling

Under the framework of GPF, the KLGPF obtains particles when the time update process is conducted. KLGPF sets an interval range by introducing Mahalanobis distance, which includes almost the range of the prior distribution of each state at any time. Then, the interval is divided into several subsections. The prior distribution is selected as an importance probability density function, from which the particles are extracted. Finally, KLGPF counts the number of particles falling into different intervals, so as to calculate the number of particles needed in real time and adjust them accordingly.

2.1 Setting the range of interval

As the mentioned above, the Mahalanobis distance is used to set the range of the interval. The Mahalanobis distance between the sample and the ensemble is defined as

$$MD(\boldsymbol{x}_i,\boldsymbol{x})\triangleq[(\boldsymbol{x}_i-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})]^{1/2} \tag{15}$$

Set the boundary value as $\boldsymbol{x}_{i(\max)}=\boldsymbol{\mu}\pm a\boldsymbol{B}\boldsymbol{b}$, and the probability of the sample that falls within the boundary at any time is $1-\theta$, where a is unknown quantity and $\boldsymbol{\Sigma}=\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}$. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and variance of the ensemble. Then

$$P((\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\leqslant(a\boldsymbol{B}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(a\boldsymbol{B}\boldsymbol{b}))=1-\theta \tag{16}$$

Since $\boldsymbol{x}$ conforms to normal distribution, Eq.(16) can be written as

$$P(\chi^2_\gamma\leqslant(a\boldsymbol{B}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(a\boldsymbol{B}\boldsymbol{b}))=1-\theta\doteq P(\chi^2_\gamma\leqslant\chi^2_{\gamma,1-\theta})=1-\theta \tag{17}$$

where $\boldsymbol{b}$ is a vector of those with the same dimension as $\boldsymbol{B}$, χ^2_γ a chi-square distribution with γ degrees of freedom, and $\chi^2_{\gamma,1-\theta}$ the $1-\theta$ quantile of the chi-square distribution. From Eq.(17), it is obtained that

$$(a\boldsymbol{B}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(a\boldsymbol{B}\boldsymbol{b})=\chi^2_{\gamma,1-\theta}\Leftrightarrow a^2\cdot\gamma=\chi^2_{\gamma,1-\theta}\Rightarrow a^2=\frac{\chi^2_{\gamma,1-\theta}}{\gamma} \tag{18}$$

According to the rightmost term of Eq.(18), a gradually decreases with the increase of γ and thus gets a maximum value of $\sqrt{\chi^2_{1,1-\theta}}$ when $\gamma=1$. To sum up, the range of interval can be obtained by substituting the boundary value into Eq.(15), which can be written as

$$\left(0,\sqrt{(a\boldsymbol{B}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(a\boldsymbol{B}\boldsymbol{b})}\right)=\left(0,\sqrt{a^2\gamma}\right)=\left(0,\sqrt{\chi^2_{1,1-\theta}\cdot\gamma}\right)$$

Therefore, a small enough θ value needs to be set to ensure that the probability of KLD-sampling

particles that falls into interval $\left(0,\sqrt{\chi^2_{1,1-\theta}\cdot\gamma}\right)$ is large enough, where γ is set to the dimension of the state vector.

2.2 Counting the number of subintervals into which particles fall

According to the selected interval range in Section 2.1, it is evenly divided into m subintervals and the size is $\Delta=\sqrt{\chi^2_{1,1-\theta}\cdot\gamma}\Big/m$, as shown in Fig.1.

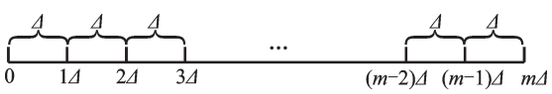

Fig.1 Interval segmentation method

An interval mapping table is presented in Table 1.

Table 1 Interval mapping table

index	0	1	2	3	⋯	m
Value	$[0,1\Delta)$	$[1\Delta,2\Delta)$	$[2\Delta,3\Delta)$	$[3\Delta,4\Delta)$	⋯	$[(m-1)\Delta,m\Delta)$

Then an interval mapping function is built by using Eq.(15) as

$$\text{index}=\left[\frac{MD(\boldsymbol{x}_n^i,\boldsymbol{x}_n)}{\Delta}+1\right]_{\text{floor}} \tag{19}$$

where $[\cdot]_{\text{floor}}$ is a floor function, and index is an index of subintervals. A detailed description of the KLGPF algorithm is given as follows.

The implementation of KLGPF:

Step 1 Initialize the particle set $\{\boldsymbol{x}_0^i\}_{i=1}^M$ based on the prior information.

Measurement update:

Step 2 Compute the weights by $\overline{w}_n^{(i)}=p(y_n|\boldsymbol{x}_n^{(i)})$ and normalize the respective weights by $w_n^{(i)}=\overline{w}_n^{(i)}\Big/\sum_{j=1}^M\overline{w}_n^{(i)}$.

Step 3 Estimate the mean and variance at the current time

$$\boldsymbol{\mu}_n=\sum_{i=1}^M w_n^{(i)}\boldsymbol{x}_n^{(i)}$$

$$\boldsymbol{\Sigma}_n=\sum_{i=1}^M w_n^{(i)}(\boldsymbol{\mu}_n-\boldsymbol{x}_n^{(i)})(\boldsymbol{\mu}_n-\boldsymbol{x}_n^{(i)})^{\mathrm{H}}$$

Time update:

Step 4 Get particles $\{\boldsymbol{x}_n^i\}_{i=1}^M$ from the posterior distribution $\mathcal{N}(\boldsymbol{x}_n;\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_n)$.

Step 5 Obtain the updated particle set $\{\boldsymbol{x}_{n+1}^i\}_{i=1}^M$ by using the process equation to update the state of each particle.

Step 6 Calculate the mean and variance of the updated particle set by

$$\boldsymbol{\mu}_{n+1}=\frac{1}{M}\sum_{i=1}^M\boldsymbol{x}_{n+1}^{(i)}$$

$$\boldsymbol{\Sigma}_{n+1}=\frac{1}{M}\sum_{i=1}^M(\boldsymbol{\mu}_{n+1}-\boldsymbol{x}_{n+1}^{(i)})(\boldsymbol{\mu}_{n+1}-\boldsymbol{x}_{n+1}^{(i)})^{\mathrm{H}}$$

KLD-sampling:

Step 7 Set the interval range as $\left(0,\sqrt{\chi^2_{1,1-\theta}\cdot\gamma}\right)$ according to the state dimension and divide the interval into m subintervals.

Step 8

(1) Draw a particle from the importance probability density function and $M=M+1$, where M describes the number of particles. At the same time, calculate the Mahalanobis distance between particles and the ensemble.

(2) If $\boldsymbol{x}_n^i$ fall into an empty subinterval, then add one to the non-empty subinterval, i.e., $k=k+1$.

(3) If $M<\frac{1}{2e}\chi^2_{k-1,1-\delta}$, then go to Step 8(1), otherwise go to Step 8(2).

3 Simulation Results

In this paper, KLGPF, KLPF, GPF, and PF algorithms are separately applied to one-dimensional nonlinear model for numerical simulation. The model is a basic univariate nonstationary growth model (UNGM), which is a strongly nonlinear model. It can be described by DSS equation[16] as

$$\begin{cases}x_n=0.5x_{n-1}+\dfrac{25x_{n-1}}{1+x_{n-1}^2}+8\cos(1.2(n-1))+Q_n & \text{Process equation}\\ y_n=\dfrac{x_k^2}{20}+R_n & \text{Observation equation}\end{cases} \tag{20}$$

where $n=1,2,\cdots,N$, $Q_n\sim\mathcal{N}(0,\sigma_Q^2)$ and $R_n\sim\mathcal{N}(0,\sigma_R^2)$ are process and measurement noises, $x_0=0$, and $N=100$ are specified. The simulation step size is set as 1. Some key parameters in KLPF are set, i.e., $e=0.15$, $\Delta=0.2$, and $\delta=0.99$. The parameters of KLGPF are set that $e=0.15$, $\delta=$

0.99, $\theta = 1 \times 10^{-9}$, and the subinterval numbers $m = [20\Sigma_n]_{\text{ceil}}$. The simulation parameters are set based on Ref.10 where Σ_n is a variance of the prior distribution at n time and $[\cdot]_{\text{ceil}}$ is a ceiling function. The statistical property of process noise and measurement noise are specified by $\sigma_Q^2 = 1, \sigma_R^2 = 1$ under normal conditions. The prior information of the initial state of the filter is $p(x_0) \sim \mathcal{N}(3, 3^2)$, and the particles of PF and GPF are the average number of KLGPF particles. In some abnormal cases, however, these noises will mutate. Therefore, it is firstly assumed that the system is under normal conditions, and then the noise is suddenly changed at some span. Fig.2 shows the true states and the estimates obtained using GPF and PF with fixed particle numbers and the estimates obtained using KLGPF and KLPF with adaptive particle numbers under normal conditions. It can be seen that the estimation of KLGPF is not much different from that of GPF, which uses the framework of GPF, and so are the KLPF and PF. The particles used by these algorithms are plotted in Fig.3. FIR result is used to display the outline information of the change in the number of particles of KLGPF. The changes in particle number of KLGPF and KLPF should also be noticed that the number of KLGPF particles changes with a priori probability, whereas that of KLPF changes with a posteriori probability. Then, KLGPF introduces the Mahalanobis distance and the size of the interval is automatically set, while KLPF does not. The time elapsed by using these algorithms are shown in Table 2. KLGPF inherits the

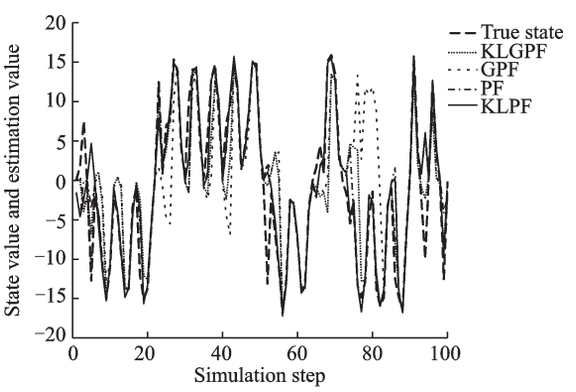

Fig.2 True state value and state estimation of KLGPF, GPF, PF, and KLPF under normal conditions

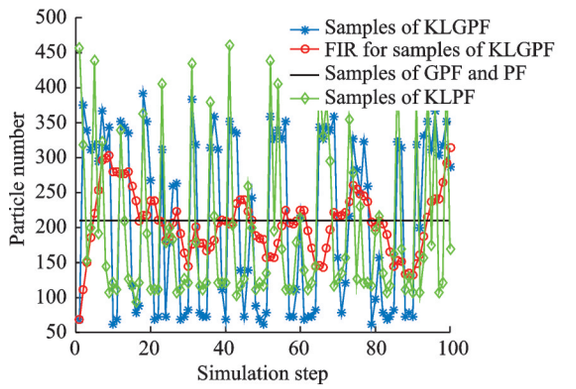

Fig.3 Changes in particle number of KLGPF, KLPF, GPF, and PF under normal conditions

Table 2 Computation time of KLGPF, KLPF, GPF, and PF under the same simulation condition

Algorithm	Time/s
KLGPF	0.115 493
KLPF	0.161 318
GPF	0.076 724
PF	0.124 189

characteristics of GPF that eliminates the process of resampling, whose complexity is $O(N)$. Since KLGPF is an algorithm that adjusts the number of the particle online in real time, the computation time increases a little compared to GPF.

Fig.4 presents the estimation results of the four approaches. The process noise is mutated to $500Q_n$ at time 20—30 and time 50—60 mutation period, which is 500 times that of normal conditions. As shown in Fig.4, KLGPF could remain stable at the mutation period, which is the same with KLPF. Fig.5 further illustrates this point that the errors of

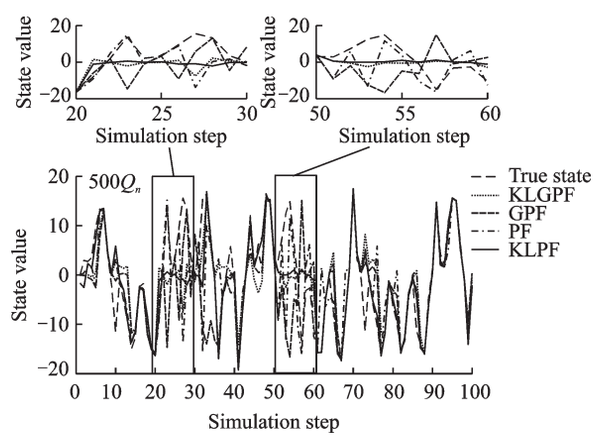

Fig.4 True state value and state estimation of KLGPF, GPF, PF, and KLPF under abnormal conditions

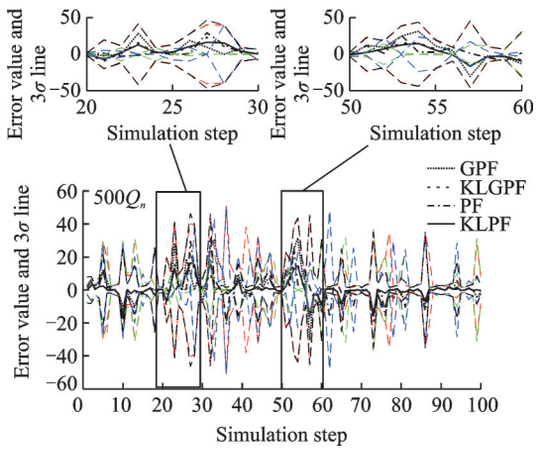

Fig.5 Estimation error and 3σ line of KLGPF, KLPF, GPF, and PF under abnormal conditions ($500Q_n$)

GPF and PF both exceed their $3\sigma_{error}$ line at mutation period. The change of particle number is shown in Fig. 6. It is obvious that the number of particles changes significantly during the mutation period, and the number of particles used by KLGPF is more than that of KLPF. However, since KLGPF is within the framework of GPF, the computation time for KLGPF is less than KLPF, as shown in Table 3.

The comparison of the root mean square error (RMSE) value of the four algorithms is presented in Fig. 7, in which the RMSE value of KLGPF is lower than those of GPF and PF. However, the performance of KLGPF in terms of RMSE value is slightly inferior to that of KLPF. This is because the RMSE value for GPF is marginally higher than that of PF. When the process noise mutates to $1\,000Q_n$ at the mutation period, both GPF and PF are divergent so that the filter could not be carried out, as shown in Fig. 8. In Table 4, the concrete average RMSE value is listed for $500Q_n$ process noise mutation and $1\,000Q_n$ process noise mutation. Similarly, the computation time is presented in Table 3.

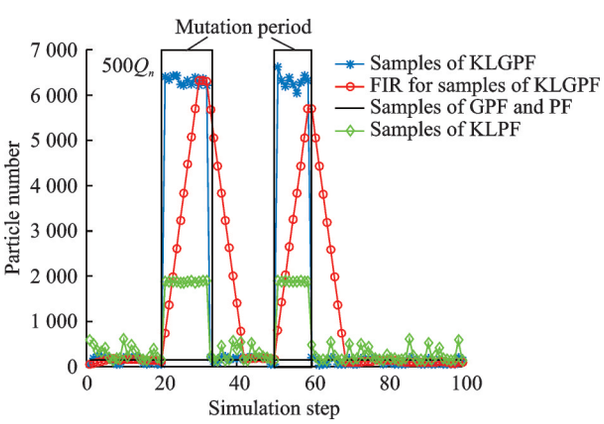

Fig.6 Changes in particle numbers of KLGPF, KLPF, GPF, and PF under abnormal conditions ($500Q_n$)

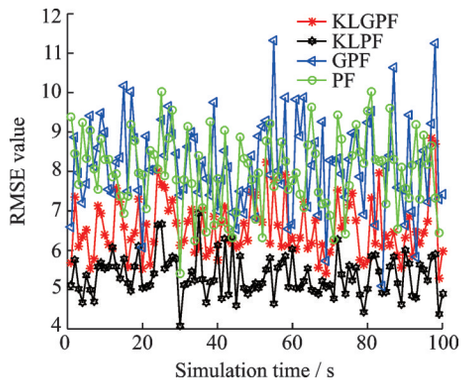

Fig.7 RMSE values of KLGPF, KLPF, GPF, and PF under abnormal conditions ($500Q_n$)

From the above information, it can be concluded that the KLGPF greatly improves the filtering speed while slightly losting its algorithm accuracy

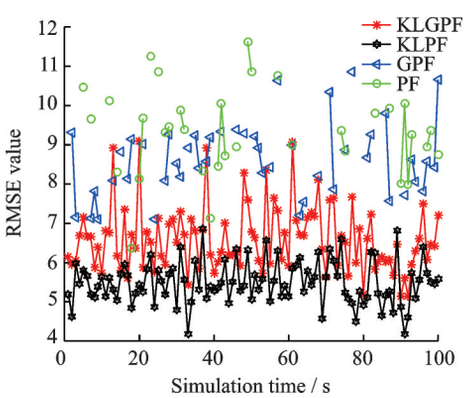

Fig.8 RMSE values of KLGPF, KLPF, GPF, and PF under abnormal conditions ($1\,000Q_n$)

Table 3 Average computation time of 100 random realizations under different abnormal conditions

Algorithm	Time/s	
	$500Q_n$	$1\,000Q_n$
KLGPF	3.170 330	8.556 011
KLPF	7.067 693	13.395 819

Table 4 Average RMSE value of 100 random realizations under different abnormal conditions

Algorithm	RMSE	
	$500Q_n$	$1\,000Q_n$
KLGPF	6.546 8	6.591 0
KLPF	5.351 3	5.488 9
GPF	8.168 8	8.635 1
PF	8.001 9	9.325 2

compared with KLPF. In Fig.9, the noise of mutation is gradually increased. KLGPF and KLPF both maintain good estimates, while the divergence frequently occurs for GPF and PF when the noise mutation value exceeds $700Q_n$.

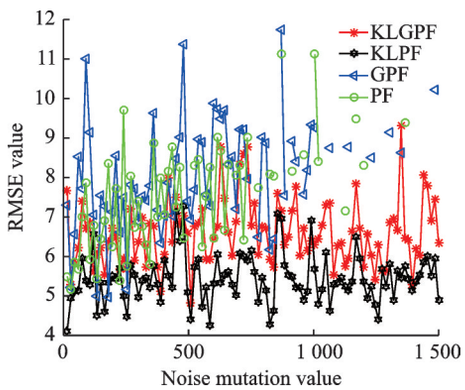

Fig.9 RMSE values of KLGPF, KLPF, GPF, and PF

4 Conclusions

KLGPF is proposed in this paper to cope with the divergence problems in the case of system noise changes. On one hand, the algorithm which combines GPF with KLD-sampling can adjust the size of particle sets in real time during the span of system noise changes. Therefore, the KLGPF can remain stable when the system is subjected to strong interference and the process noise experiences sudden changes.

On the other hand, the speed of KLGPF is much faster than that of KLPF, albeit the RMSE value of KLGPF is marginally higher. This is because the improved algorithm has the feature of GPF that it does not require the systematic resampling procedure with complexity $O(N)$. The predictive distribution is approximated by Gaussian distribution in KLGPF, which results in higher RMSE value than KLPF. Moreover, KLGPF introduces the Mahalanobis distance to set the interval that is independent of the original data unit, so the KLGPF can be easily combined with other applications.

References

[1] ZONG Qun, LIU Penghao, DONG Qi, et al. Multi-robot object recognition and formation tracking based on point cloud covariance descriptors[J]. Journal of Tianjin University (Science and Technology), 2017, 50(11): 1160-1168. (in Chinese)

[2] CARMI A, OSHMAN Y. Adaptive particle filtering for spacecraft attitude estimation from vector observations[J]. Journal of Guidance, Control, and Dynamics, 2009, 32(1): 232-241.

[3] BRUNO M G S. Sequential Monte Carlo methods for nonlinear discrete-time filtering[J]. Synthesis Lectures on Signal Processing, 2013, 6(1): 99.

[4] SILVA I R. Confidence intervals through sequential Monte Carlo[J]. Computational Statistics & Data Analysis, 2017, 105(1): 112-124.

[5] KOTECHA J H, DJURIC P M. Gaussian particle filtering[C]//Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing. Piscataway: IEEE, 2003: 429-432.

[6] CHEN Peng, QIAN Hui, ZHU Miaoliang. A fast Gaussian particle filtering algorithm[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2008, 36(S1): 291-294. (in Chinese)

[7] YANG Y B, LIANG Y, PAN Q, et al. Adaptive Gaussian mixture filter for Markovian jump nonlinear systems with colored measurement noises[J]. ISA Transactions, 2018, 80(9): 111-126.

[8] HUEMMER C, HOFMANN C, MAAS R, et al. Estimating parameters of nonlinear systems using the elitist particle filter based on evolutionary strategies[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, 26(3): 595-608.

[9] AUDELIANO W L, GUILHERME S B A. Hybrid self-adaptive particle filter through KLD-sampling and SAMCL[C]//Proceedings of the 18th International Conference on Advanced Robotics. [S.l.]: IEEE, 2017: 106-111.

[10] FOX D. KLD-sampling: Adaptive particle filters[J]. Advances in Neural Information Processing Systems, 2001, 14(1): 713-720.

[11] SUN D H, QIN H, ZHAO M, et al. Adaptive KLD sampling based Monte Carlo localization[C]//Proceedings of 2018 Chinese Control and Decision Conference (CCDC). Shenyang: IEEE, 2018: 4154-4159.

[12] LI T, SUN S, SATTAR T P. Adapting sample size in particle filters through KLD-resampling[J]. Electronics Letters, 2013, 49(12): 740-742.

[13] STRAKA O, ŠIMANDL M. Sample size adaptation for particle filters[J]. IFAC Proceedings Volumes,

2004, 37(6): 437-442.

[14] DOUCET A, PITT M K, DELIGIANNIDIS G, et al. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator[J]. Biometrika, 2015, 102(2): 295-313.

[15] RAMOS P L, ALMEIDA M P, TOMAZELLA V L D, et al. Improved Bayes estimators and prediction for the Wilson-Hilferty distribution[J] Anais da Academia Brasileira de Ciencias, 2019. DOI: 10.1590/0001-3765201920190002

[16] KOTECHA J H, DJURIC P M. Gaussian sum particle filtering[J]. IEEE Transactions on Signal Processing, 2003, 51(10): 2602-2612.

Acknowledgements This work is supported by the China Postdoctoral Science Foundation(No.171980), the National Natural Science Foundation of China (Nos. 61973160, 51505221) and Key Laboratory Fund of Science and Technology on Communication Networks(No. 6142104180114).

Author Dr. **ZHOU Zhaihe** received the M.S. and Ph.D. degrees in navigation, guidance and control from Nanjing University of Aeronautics and Astronautics (NUAA) in 2002 and 2010, respectively. He is currently an associate professor with College of Automation Engineering, NUAA. His main research directions are navigation positioning and data fusion technology, embedded system and electronic technology.

Author contributions Dr. **ZHOU Zhaihe** conducted the progress of this paper. Mr. **ZHONG Yulu** wrote the manuscript. Dr. **ZENG Qingxi** contributed to revising and submitting this paper. Dr. **TIAN Xiangrui** contributed to translating the manuscript.

Competing interests The authors declare no competing interests.

(Production Editor: WANG Jing)

基于KLD采样改进的高斯粒子滤波算法

周翟和, 钟雨露, 曾庆喜, 田祥瑞

(南京航空航天大学自动化学院,南京 211106,中国)

摘要:为了自适应地调整滤波样本,本文提出了一种基于Kullback-Leibler散度(Kullback-Leible divergence, KLD)-抽样的改进高斯粒子滤波算法(Gaussian particle filter algorithm based on KLD, KLGPF)。在采样过程中,算法通过计算KLD来调整粒子集的大小,使其介于粒子的离散概率密度函数和真实的后验概率密度函数之间。当噪声服从高斯分布,且噪声的统计特性发生突变时,KLGPF具有显著的效果,仿真结果表明,KLGPF在噪声统计量突变时仍能保持良好的估计效果。在相同条件下,KLGPF的运算速度相比基于KLD采样的粒子滤波算法(Particle filter algorithm based on KLD,KLPF)的运算速度提高了28%。

关键词:粒子滤波;高斯粒子滤波;KLD采样;噪声突变;自适应粒子数