sensors

MDPI

Article

An Overdispersed Black-Box Variational Bayesian–Kalman Filter with Inaccurate Noise Second-Order Statistics

Lin Cao [1,2], Chuyuan Zhang [1,2], Zongmin Zhao [1,2,*], Dongfeng Wang [3], Kangning Du [1,2], Chong Fu [4] and Jianfeng Gu [5,*]

[1] Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, Beijing Information Science and Technology University, Beijing 100101, China; charlin26@163.com (L.C.); zhangchuyuan2012@163.com (C.Z.); kangningdu@bistu.edu.cn (K.D.)
[2] School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China
[3] Beijing TransMicrowave Technology Company, Beijing 100080, China; wdf@tsmtc.com
[4] Department of Communication and Electronics Engineering, School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China; fuchong@mail.neu.edu.cn
[5] Moonshot Health, 3700 St-Patrick Street, Suite 102, Montreal, QC H4E 1A2, Canada
* Correspondence: zhaozongmin@bistu.edu.cn (Z.Z.); jianfenggu@hotmail.com (J.G.)

Abstract: Aimed at the problems in which the performance of filters derived from a hypothetical model will decline or the cost of time of the filters derived from a posterior model will increase when prior knowledge and second-order statistics of noise are uncertain, a new filter is proposed. In this paper, a Bayesian robust Kalman filter based on posterior noise statistics (KFPNS) is derived, and the recursive equations of this filter are very similar to that of the classical algorithm. Note that the posterior noise distributions are approximated by overdispersed black-box variational inference (O-BBVI). More precisely, we introduce an overdispersed distribution to push more probability density to the tails of variational distribution and incorporated the idea of importance sampling into two strategies of control variates and Rao–Blackwellization in order to reduce the variance of estimators. As a result, the convergence process will speed up. From the simulations, we can observe that the proposed filter has good performance for the model with uncertain noise. Moreover, we verify the proposed algorithm by using a practical multiple-input multiple-output (MIMO) radar system.

Keywords: MIMO radar; Kalman filter; Bayesian robustness; uncertain noise; posterior noise statistics; proposal distribution; variational inference

check for updates

Citation: Cao, L.; Zhang, C.; Zhao, Z.; Wang, D.; Du, K.; Fu, C.; Gu, J. An Overdispersed Black-Box Variational Bayesian–Kalman Filter with Inaccurate Noise Second-Order Statistics. *Sensors* **2021**, *21*, 7673. https://doi.org/10.3390/s21227673

Academic Editors: Shiyang Tang, Zhanye Chen, Yan Huang and Ping Guo

Received: 23 September 2021
Accepted: 16 November 2021
Published: 18 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recently, the design of robust filters has been one of the most popular topics in modern radar systems and MIMO-based radar system in particular. There are two main reasons why this has occurred. First, a radar signal of interest can be corrupted naturally from other interference and noise or by clutter and so on, which results in uncertainty in assumed statistical models of given received signals. Second, the models of measurement system can be affected by the setup of transmitters and receivers in practical applications of radar systems. Unfortunately, it is impossible or prohibitively expensive to design an optimal filter by obtaining and/or understanding accurate models in the real world, which makes some nominally optimal filters suffer significant degradations in performance even if small deviations from the assumed models occur. Therefore, a robust filter can be considered as a kind of optimal filter under some conditions of uncertainty models. The seminal concept of robust filters is proposed in the 1970s [1,2], where the minimax approach was discussed for the Wiener filter, which is a batch-based optimal filter for stationary signals. Currently, the Bayesian-based robust approach has attracted significant attention in the design of robust filters due to the use of information with respect to the prior distribution of models in providing more accurate knowledge on the statistical model.

It is known to all that classical Kalman filtering [3] involves an optimal filter in terms of the Gaussian and linear assumptions of dynamic systems. It has been well established in many fields such as communication, navigation, radar and control. In addition, more and more Kalman filtering-related technologies are applied to the localization and tracking of moving agents [4–6]. However, it also suffers from prominent shortcomings, for example, the good performance lies in completely grasping the statistical models of noise and the state of signal. As a result, the problem of designing a robust Kalman filter in practical applications where the knowledge of noise distribution is missing or imprecise is a big challenge for both researchers and developers. The authors of [7–12] proposed methods, which include so-called adaptive Kalman filtering, to estimate signal states and noise simultaneously, which works well when there is a lot of data used to obtain certain accurate performances in the entire estimated period. Later on, minimax-based Kalman filters [13–15] and finite impulse response Kalman filters [16–19] were presented.

Inspired by both the Bayesian innovation process and the Bayesian orthogonality principle, an intrinsically Bayesian robust Kalman filter (IBRKF) was proposed in [20]. The authors argued that it should be optimal from the point of view of the cost function and the prior distribution of uncertain noise. Herein, its recursive equations are very similar to that of the classical Kalman filter. The IBRKF makes use of concepts of effective noise statistics and effective Kalman gain by replacing the corresponding variables in the classical Kalman filter framework in order to improve performance. Therefore, IBRKF can be considered as a robust Kalman filter in terms of uncertain noise.

The IBR method can achieve optimal performances in terms of the uncertainty of models by using prior distributions, which does not concern measurements, and may contain some useful information for improving performances further. Instead, the optimal Bayesian Kalman filter (OBKF) takes into account the observation sequence and prior knowledge simultaneously in order to achieve optimal filtering [21]. It designs a factor-graph-based message passing algorithm, which transforms the global function into a form of local function multiplication. In this manner, it integrates the effective information of each local node and lays the foundation for approximating posterior noise distributions. Moreover, note that the recursive structure of OBKF is also the same as that of the classical Kalman filter. However, the filtering process of OBKF is quite complicated. It needs to pass messages from the beginning when calculating posterior parameters at each moment. When the dimension of the matrix or observation sequence is large, the time cost of OBKF is catastrophic. In addition, the maximum a posteriori criterion and variational inference are common methods for estimating the unknown noise covariance matrix [22–26]. In terms of performance, although these methods outperform classical filtering algorithms, there is still room for improvement. For example, Refs. [22,23] only solves the problem of the unknown observation noise covariance matrix while ignoring process noise. Coincidentally, Ref. [24] used the variational Bayesian method to estimate the system state and observation noise while simultaneously using the maximum a posteriori method to estimate process noise. By selecting the inverse Wishart distribution as the conjugate prior for the covariance matrix of a Gaussian distribution with known mean, Refs. [25,26] jointly estimated the state vector, prediction error and observation noise covariance matrix under the framework of variational inference. It is undeniable that these methods increased the complexity of unknown noise covariance matrix estimation. Therefore, the aim of our study is to improve the real-time performance of the algorithm as much as possible under the premise of ensuring performance.

Thanks to the pioneering work of Bode, Shannon, etc., the concept of innovation process is widely used in the derivation of Kalman recursive equations [27–29]. In this paper, the recursive equations of KFPNS are derived based on Bayesian theory. Compared with the recursive equations of a classical Kalman filter, it only needs to calculate the posterior noise statistics, which are obtained by variational distribution. The variational distribution is an approximate distribution of the real posterior noise distribution. Firstly, we joined the observation sequence and the prior distribution of uncertain noise parameters based on the Markov hypothesis, and then we used the O-BBVI method to approximate real posterior noise distributions. Herein, we assume that the state-space model is Gaussian. The posterior noise statistics are the posterior expectation of the second-order noise statistics and are also one of the parameters in the process of approximation. In other words, we extend the IBRKF framework to an a posteriori version based on an observation sequence and design a new approach to solve the problem of uncertain noise parameters.

Our KFPNS is universal. Any model in which the noise second-order statistics can be assumed to be a certain probability distribution is suitable for this framework. However, note that under the assumption of Gaussian noise, the proposed method can obtain better estimation accuracy, while non-Gaussian noise will produce larger errors. Moreover, the run time of KFPNS is a difficult problem when the dimension of observation sequence is large, because the time cost in computing the posterior noise statistics mainly depends on the number of samples and the length of observation sequence. Fortunately, if the results of multiple calculations of the posterior noise statistics change little, we can consider skipping this step.

2. Materials and Methods

Let us first introduce some notations to be used in this paper. Lowercase bold letters represent vectors and uppercase bold letters represent matrices, respectively. We use $\mathbf{x}_k$ to represent the value of $\mathbf{x}$ at time k. $\mathbb{E}[\cdot]$ is used to denote the expectation of the random vector. $\mathbf{A}^T$ stands for a transpose operator. $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ represents a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

2.1. Kalman Filter

The authors of [3] proposed a discrete time-domain filtering scheme in the linear state-space model, which is optimal for all linear system models. The state-space model usually consists of the following equations.

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{\Gamma}_k \mathbf{u}_k, \tag{1}$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \tag{2}$$

Among them, $\mathbf{x}_k$ and $\mathbf{y}_k$ are the $n \times 1$ dimensional state vector and the $m \times 1$ dimensional observation vector, respectively. Both $\mathbf{u}_k$ and $\mathbf{v}_k$ are zero-mean random vectors, called process noise and observation noise, and their sizes are $s \times 1$ and $m \times 1$, respectively. $\mathbf{\Phi}_k$ is the state transition matrix of $n \times n$. $\mathbf{\Gamma}_k$ represents the process noise transition matrix of $n \times s$. $\mathbf{H}_k$ is an $m \times n$ matrix called the observation transition matrix. The statistics of the above vector are as follows:

$$\begin{array}{l} \mathbb{E}\left[\mathbf{u}_k \mathbf{u}_l^T\right] = \mathbf{Q}\delta_{kl}, \mathbb{E}\left[\mathbf{v}_k \mathbf{v}_l^T\right] = \mathbf{R}\delta_{kl} \ \forall k,l = 0,1,\ldots \\ \mathbb{E}\left[\mathbf{v}_k \mathbf{x}_l^T\right] = 0_{m\times n}, \mathbb{E}\left[\mathbf{u}_k \mathbf{v}_l^T\right] = 0_{s\times m} \ \forall k,l = 0,1,\ldots \\ \mathbb{E}\left[\mathbf{u}_k \mathbf{y}_l^T\right] = 0_{s\times m}, \ 0 \le l \le k, \end{array} \tag{3}$$

where $0_{m\times n}$ is a zero-valued matrix of size $m \times n$. The purpose of Kalman filtering is to obtain the least square estimate $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$ from observation $\mathbf{y}_l (l \le k-1)$.

2.2. Introduction to Bayesian Robust Filters

In engineering, the Bayesian criterion and minimax criterion (instead of the Bayesian method when there is no prior knowledge) are usually used to design robust filters [30–32]. The meaning of a Bayesian criterion is as follows: The average cost is the least. In the context of state estimation, the average cost is generally reflected by the mean-square error (MSE); in other words, we have the following.

$$C(\mathbf{x}_k, \psi(\mathbf{y}_k)) = \mathbb{E}\left[(\mathbf{x}_k - \psi(\mathbf{y}_k))^T (\mathbf{x}_k - \psi(\mathbf{y}_k))\right]. \tag{4}$$

According to the Bayesian criterion, we minimize the cost function to find an optimal filter:

$$\hat{\psi}(\mathbf{y}_k) = \arg\min_{\psi \in \Psi} C(\mathbf{x}_k, \psi(\mathbf{y}_k)), \tag{5}$$

where Ψ is the possible class of filters. The filter that satisfies the above formula is the minimum mean square error filter.

When we cannot obtain all the knowledge of the model, we assume that the model is controlled by an unknown parameter $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_l]$, $\boldsymbol{\theta} \in \Theta$, where Θ is the set of all possible parameters, called the uncertain class. Fortunately, these parameters can be obtained by solving the corresponding posterior distribution; thus, the Bayesian robust filter can be expressed as follows:

$$\psi^{\Theta}(\mathbf{y}_k) = \arg\min_{\psi \in \Psi} \mathbb{E}_{\theta}[\, C_{\theta}(\mathbf{x}_k, \psi(\mathbf{y}_k)) \,|\mathcal{Y}_k], \tag{6}$$

where the expectation is taken relative to the posterior distribution $p(\,\boldsymbol{\theta}|\,\mathcal{Y}_k)$, and $C_{\boldsymbol{\theta}}(\cdot)$ is the cost function relative to the unknown parameter $\boldsymbol{\theta}$. Dehghannasiri proposed that the average cost of the filter relative to the uncertain class is equal to the error of applying the filter to the effective information [20]. Therefore, the process of designing a robust filter for an uncertain model can be transformed into designing a Bayesian robust filter using posterior effective information.

2.3. Bayesian Robust Kalman Filter Based on Posterior Noise Statistics

2.3.1. KFPNS Framework

Now assume that the second-order statistical knowledge of process noise and observation noise is unknown and that it is determined by the unknown parameters α_1 and α_2. Here, the covariance matrices of these two noises are as follows.

$$\mathbb{E}\left[\mathbf{u}_k^{\alpha_1}\left(\mathbf{u}_l^{\alpha_1}\right)^T\right] = \mathbf{Q}^{\alpha_1}\delta_{kl}, \tag{7}$$

$$\mathbb{E}\left[\mathbf{v}_k^{\alpha_2}\left(\mathbf{v}_l^{\alpha_2}\right)^T\right] = \mathbf{R}^{\alpha_2}\delta_{kl}. \tag{8}$$

If α_1 and α_2 are statistically independent, then $p(\boldsymbol{\alpha}) = p(\alpha_1)p(\alpha_2)$ stands for the prior distribution. The state-space model can be parameterized as follows.

$$\mathbf{x}_{k+1}^{\alpha_1} = \mathbf{\Phi}_k \mathbf{x}_k^{\alpha_1} + \mathbf{\Gamma}_k \mathbf{u}_k^{\alpha_1}, \tag{9}$$

$$\mathbf{y}_k^{\boldsymbol{\alpha}} = \mathbf{H}_k \mathbf{x}_k^{\alpha_1} + \mathbf{v}_k^{\alpha_2}. \tag{10}$$

Although the IBRKF is optimal in the sense of prior, there is still a certain deviation between the prior expectation of noise and the actual noise, which makes it impossible to achieve real optimal filtering. Therefore, this paper focuses on the influence of posterior noise distribution on the cost function. With $\mathcal{Y}_k = \{\mathbf{y}_0, \ldots, \mathbf{y}_k\}$ as the observation sequence and analysis based on (6), the linear filtering function that satisfies the definition of a Bayesian robust filter is as follows:

$$\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}} = \sum_{l \le k-1} \mathbf{\Omega}_{k,l}^{\Theta} \mathbf{y}_l^{\boldsymbol{\alpha}}, \tag{11}$$

such that the following is the case:

$$\mathbf{\Omega}_{k,l}^{\Theta} = \arg\min_{\mathbf{\Omega}_{k,l} \in \Im} \mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\left(\mathbf{x}_k^{\alpha_1} - \sum_{l \le k-1} \mathbf{\Omega}_{k,l}\mathbf{y}_l^{\boldsymbol{\alpha}}\right)^T \left(\mathbf{x}_k^{\alpha_1} - \sum_{l \le k-1} \mathbf{\Omega}_{k,l}\mathbf{y}_l^{\boldsymbol{\alpha}}\right)\right] \middle| \mathcal{Y}_{k-1}\right], \tag{12}$$

where $\Im$ is the vector space of all $n \times m$ matrix-valued functions, and $\mathbf{\Omega}_{k,l} \in \Im$ is a mapping $\mathbf{\Omega}_{k,l} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^{n \times m}$ such that $\sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \|\mathbf{\Omega}_{k,l}\|_2 < \infty$, $\|\cdot\|_2$ being the L_2 norm.

The derivation of KFPNS recursive equations depends on the following theorem, definition and lemma. Our equations are only a posterior version relative to [20], and the proof of related theories are similar to that in [20].

Theorem 1. *(Bayesian Orthogonality Principle) If the weight function $\boldsymbol{\Omega}_{k,l}^{\Theta}$ of a linear filter satisfies (12), then the $\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$ obtained in (11) is called the optimal Bayesian least-squares estimation if and only if the following is the case.*

$$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\left(\mathbf{x}_k^{\alpha_1}-\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}\right)\left(\mathbf{y}_l^{\boldsymbol{\alpha}}\right)^T\right]\,\middle|\mathcal{Y}_{k-1}\right]=0_{n\times m}\;\forall l\leq k-1. \tag{13}$$

Definition 1. *(Bayesian innovation process) According to the state-space model described in (9) and (10), if $\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$ is the least-square estimate of $\mathbf{x}_k^{\alpha_1}$, then the stochastic process of the following:*

$$\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}=\mathbf{y}_k^{\boldsymbol{\alpha}}-\mathbf{H}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}} \tag{14}$$

is a zero-mean process and $\forall l,l'\leq k-1$; the following is obtained.

$$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\widetilde{\mathbf{z}}_l^{\boldsymbol{\alpha}}\left(\widetilde{\mathbf{z}}_{l'}^{\boldsymbol{\alpha}}\right)^T\right]\,\middle|\mathcal{Y}_{k-1}\right]=\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{H}_l\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_l^T+\mathbf{R}^{\alpha_2}\middle|\mathcal{Y}_{k-1}\right]\delta_{ll'}. \tag{15}$$

Lemma 1. *(Bayesian information equivalence) The Bayesian least-squares estimation obtained based on $\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}$ is equivalent to the Bayesian least-squares estimation calculated based on $\mathbf{y}_k^{\boldsymbol{\alpha}}$; in other words, we have the following.*

$$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\left(\mathbf{x}_k^{\alpha_1}-\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}\right)\left(\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}\right)^T\right]\,\middle|\mathcal{Y}_{k-1}\right]=0_{n\times m}. \tag{16}$$

Then, (11) can be rewritten as follows.

$$\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}=\sum_{l\leq k-1}\boldsymbol{\Omega}_{k,l}^{\Theta}\widetilde{\mathbf{z}}_l^{\boldsymbol{\alpha}}. \tag{17}$$

By substituting it into (16), after moving the terms on the left and right sides of the equation, we obtain the following.

$$\boldsymbol{\Omega}_{k,l}^{\Theta}=\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\mathbf{x}_k^{\alpha_1}\left(\widetilde{\mathbf{z}}_l^{\boldsymbol{\alpha}}\right)^T\right]\middle|\mathcal{Y}_{k-1}\right]\mathbb{E}_{\boldsymbol{\alpha}}{}^{-1}\left[\mathbf{H}_l\mathbf{P}_l^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_l^T+\mathbf{R}^{\alpha_2}\middle|\mathcal{Y}_{k-1}\right]. \tag{18}$$

In this case, the linear function of Bayesian robust filter based on posterior noise statistics is as follows.

$$\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}=\sum_{l\leq k-1}\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbb{E}\left[\mathbf{x}_k^{\alpha_1}\left(\widetilde{\mathbf{z}}_l^{\boldsymbol{\alpha}}\right)^T\right]\,\middle|\mathcal{Y}_{k-1}\right]\mathbb{E}_{\boldsymbol{\alpha}}{}^{-1}\left[\mathbf{H}_l\mathbf{P}_l^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_l^T+\mathbf{R}^{\alpha_2}\middle|\mathcal{Y}_{k-1}\right]\widetilde{\mathbf{z}}_l^{\boldsymbol{\alpha}}. \tag{19}$$

Thus, the state update equation is as follows:

$$\hat{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}}=\boldsymbol{\Phi}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}+\boldsymbol{\Phi}_k\mathbf{K}_k^{\Theta}\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}, \tag{20}$$

where the following is the case.

$$\mathbf{K}_k^{\Theta'}=\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]\mathbf{H}_k^T\mathbb{E}_{\boldsymbol{\alpha}}{}^{-1}\left[\mathbf{H}_k\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_k^T+\mathbf{R}^{\alpha_2}\middle|\mathcal{Y}_{k-1}\right]. \tag{21}$$

Next, let $\widetilde{\mathbf{x}}_k^{\boldsymbol{\alpha}}=\mathbf{x}_k^{\alpha_1}-\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$ denote the estimation error at time k. After some mathematical operations, $\widetilde{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}}$ can be expanded into the following.

$$\widetilde{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}}=\boldsymbol{\Phi}_k\left(\mathbf{I}-\mathbf{K}_k^{\Theta'}\mathbf{H}_k\right)\widetilde{\mathbf{x}}_k^{\boldsymbol{\alpha}}+\boldsymbol{\Gamma}_k\mathbf{u}_k^{\alpha_1}-\boldsymbol{\Phi}_k\mathbf{K}_k^{\Theta'}\mathbf{v}_k^{\alpha_2}. \tag{22}$$

We can find the update equation of the estimation error covariance matrix from this equation, which is described as follows.

$$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_{k+1}^{\mathbf{x},\boldsymbol{\alpha}}\middle|\mathcal{Y}_k\right] = \boldsymbol{\Phi}_k\left(\mathbf{I}-\mathbf{K}_k^{\Theta'}\mathbf{H}_k\right)\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]\boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]\boldsymbol{\Gamma}_k^T. \tag{23}$$

This completes the construction of KFPNS. Note that in (21) and (23), the observation sequences corresponding to the estimation error covariance matrix are different, which are $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]$ and $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]$, respectively, while the one generated by the last iteration of the recursive equation should be $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]$. For this problem, we can iterate from the beginning; that is, first calculate $\mathbf{K}_0^{\Theta'}$ with $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_0^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{-1}\right]$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k]$, then use $\mathbf{K}_0^{\Theta'}$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]$ to compute $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_1^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]$ and repeat the process until we obtain $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]$. However, this strategy requires reiteration from the origin after updating posterior noise statistics, which makes the algorithm heavily loaded. In order to improve the efficiency of the algorithm, we assume that $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right] \approx \mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]$.

Table 1 reports the difference between these two recursive equations. KFPNS replaces $\mathbf{K}_k$, $\mathbf{P}_k^x$, $\mathbf{Q}$ and $\mathbf{R}$ in the classical Kalman filter with $\mathbf{K}_k^{\Theta'}$, $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]$, $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k]$, but their recursive structures are the same. Figure 1 is the overall framework of KFPNS. The key is that KFPNS updates the state based on posterior noise statistics. Therefore, KFPNS is an improved version of the classical Kalman filter concerning posterior noise distribution.

Table 1. Comparison of the recursive equations for the classical algorithm and KFPNS.

Classical Kalman Filter	KFPNS
$\widetilde{\mathbf{z}}_k = \mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k$	$\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}} = \mathbf{y}_k^{\boldsymbol{\alpha}} - \mathbf{H}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$
$\mathbf{K}_k = \mathbf{P}_k^{\mathbf{x}}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^{\mathbf{x}}\mathbf{H}_k^T+\mathbf{R}\right)^{-1}$	$\mathbf{K}_k^{\Theta'} = \mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]\mathbf{H}_k^T\mathbb{E}_{\boldsymbol{\alpha}}{}^{-1}\left[\mathbf{H}_k\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_k^T+\mathbf{R}^{\alpha_2}|\mathcal{Y}_{k-1}\right]$
$\hat{\mathbf{x}}_{k+1} = \boldsymbol{\Phi}_k\hat{\mathbf{x}}_k + \boldsymbol{\Phi}_k\mathbf{K}_k\widetilde{\mathbf{z}}_k$	$\hat{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}} = \boldsymbol{\Phi}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}} + \boldsymbol{\Phi}_k\mathbf{K}_k^{\Theta}\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}$
$\mathbf{P}_{k+1}^{\mathbf{x}} = \boldsymbol{\Phi}_k(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{\mathbf{x}}\boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k\mathbf{Q}\boldsymbol{\Gamma}_k^T$	$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_{k+1}^{\mathbf{x},\boldsymbol{\alpha}}\middle|\mathcal{Y}_k\right] = \boldsymbol{\Phi}_k\left(\mathbf{I}-\mathbf{K}_k^{\Theta'}\mathbf{H}_k\right)\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]\boldsymbol{\Phi}_k^T$ $+\boldsymbol{\Gamma}_k\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]\boldsymbol{\Gamma}_k^T$

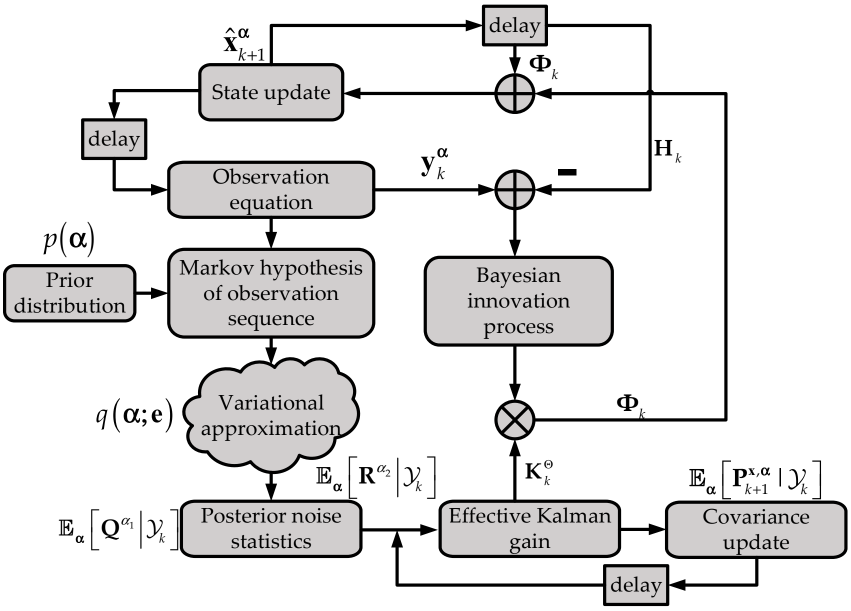

$\hat{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}}$
delay
$\boldsymbol{\Phi}_k$
State update
delay
$\mathbf{H}_k$
$\mathbf{y}_k^{\boldsymbol{\alpha}}$
-
Observation equation
$p(\boldsymbol{\alpha})$
Prior distribution
Markov hypothesis of observation sequence
Bayesian innovation process
$q(\boldsymbol{\alpha};\mathbf{e})$
Variational approximation
$\boldsymbol{\Phi}_k$
$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{R}^{\alpha_2}\middle|\mathcal{Y}_k\right]$
$\mathbf{K}_k^{\Theta}$
$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_{k+1}^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]$
$\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{Q}^{\alpha_1}\middle|\mathcal{Y}_k\right]$
Posterior noise statistics
Effective Kalman gain
Covariance update
delay

Figure 1. KFPNS framework.

2.3.2. The Calculation Method of Posterior Noise Statistics

We need to additionally calculate $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k]$ by replacing the fixed preset values, and the core process of obtaining these two conditional expectations lies in the posterior distribution $p(\boldsymbol{\alpha}|\mathcal{Y}_k)$, which is unknown. Therefore, the goal is to solve the probability density function of the unknown parameter $\boldsymbol{\alpha}$ under the condition of given observation sequence $\mathcal{Y}_k$. Although $p(\boldsymbol{\alpha}|\mathcal{Y}_k) \propto f(\mathcal{Y}_k|\boldsymbol{\alpha})p(\boldsymbol{\alpha})$, it is difficult to directly acquire a closed-form solution by relying on the prior distribution. However, a known simple distribution can be used to approximate a complex distribution, and by limiting the type of approximate distribution, a locally optimal approximate posterior distribution can be achieved. Unless otherwise specified, the distributions presented in this paper are Gaussian.

Consider a simple variational distribution, which belongs to the following exponential family:

$$q(\boldsymbol{\alpha};\mathbf{e}) = g(\boldsymbol{\alpha})\exp\Big\{\mathbf{e}^T t(\boldsymbol{\alpha}) - A(\mathbf{e})\Big\}, \tag{24}$$

where $g(\boldsymbol{\alpha})$ is the base measure, $\mathbf{e}$ is the natural parameter, $t(\boldsymbol{\alpha})$ is the sufficient statistics and $A(\mathbf{e})$ is the log-normalizer. Based on the considerations of this paper, we believe that $\mathbf{e}$ represents the expectations, $t(\boldsymbol{\alpha})$ can completely characterize all quantities of a probability distribution and $t(\boldsymbol{\alpha}) = \boldsymbol{\alpha}$. Take a simple multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\Lambda})$ as an example ($\boldsymbol{\Lambda}$ represents the identity matrix) and transform it into an exponential family form of the following:

$$\begin{aligned} p(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\Lambda}) &= \tfrac{1}{(2\pi)^{d/2}|\boldsymbol{\Lambda}|^{1/2}}\exp\Big\{-\tfrac{1}{2}(\boldsymbol{\alpha}-\mathbf{e})^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{\alpha}-\mathbf{e})\Big\} \\ &= (2\pi)^{-d/2}|\boldsymbol{\Lambda}|^{1/2}\exp\Big\{\mathbf{e}^T\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{\Lambda}\boldsymbol{\alpha} - \tfrac{1}{2}\mathbf{e}^T\boldsymbol{\Lambda}\mathbf{e}\Big\} \\ &= (2\pi)^{-d/2}|\boldsymbol{\Lambda}|^{1/2}\exp\Big\{-\tfrac{1}{2}\boldsymbol{\Lambda}\boldsymbol{\alpha}^T\boldsymbol{\alpha}\Big\}\exp\Big\{\mathbf{e}^T\boldsymbol{\alpha} - \tfrac{1}{2}\boldsymbol{\Lambda}\mathbf{e}^T\mathbf{e}\Big\}, \end{aligned} \tag{25}$$

thus, the following is obtained.

$$\begin{gathered} g(\boldsymbol{\alpha}) = (2\pi)^{-d/2}|\boldsymbol{\Lambda}|^{1/2}\exp\Big\{-\tfrac{1}{2}\boldsymbol{\Lambda}\boldsymbol{\alpha}^T\boldsymbol{\alpha}\Big\}, \\ t(\boldsymbol{\alpha}) = \boldsymbol{\alpha}, \\ A(\mathbf{e}) = \tfrac{1}{2}\boldsymbol{\Lambda}\mathbf{e}^T\mathbf{e}. \end{gathered} \tag{26}$$

In addition, the exponential family distribution also has the following property.

$$\nabla_{\mathbf{e}}A(\mathbf{e}) = \mathbb{E}_{q(\boldsymbol{\alpha};\mathbf{e})}[t(\boldsymbol{\alpha})]. \tag{27}$$

By using this property, the mutual transformation between expectation calculation and derivation calculation can be completed in variational inference. Therefore, most applications of variational inference use exponential family functions. More importantly, based on this simple property, we use the strategy of control variates to reduce the variance of the Monte Carlo gradient. At the same time, in this paper, we hope to use $q(\boldsymbol{\alpha};\mathbf{e})$ to approximate $p(\boldsymbol{\alpha}|\mathbf{y}_k)$, and the special form of exponential family distributions renders the approximation process traceable. More precisely, we consider $\boldsymbol{\alpha}$ as a Gaussian distribution, which belongs to the exponential family, and its natural parameter $\mathbf{e}$ represents expectations. Therefore, the process of finding the optimal natural parameters that makes $q(\boldsymbol{\alpha};\mathbf{e})$ the closest to $p(\boldsymbol{\alpha}|\mathbf{y}_k)$ involves calculating posterior noise statistics.

Now, suppose $q(\boldsymbol{\alpha};\mathbf{e})$ can be decomposed into the following.

$$q(\boldsymbol{\alpha};\mathbf{e}) = q(\alpha_1;e_1)q(\alpha_2;e_2). \tag{28}$$

Based on the idea of variational inference, we try to minimize Kullback–Leibler divergence $D_{KL}[q(\boldsymbol{\alpha};\mathbf{e})\|p(\boldsymbol{\alpha}|\mathcal{Y}_k)]$ so that $q(\boldsymbol{\alpha};\mathbf{e})$ approaches $p(\boldsymbol{\alpha}|\mathcal{Y}_k)$. This idea still cannot be realized directly, because D_{KL} contains an unknown distribution $p(\boldsymbol{\alpha}|\mathcal{Y}_k)$. Fortunately, after some logarithmic operations, minimizing D_{KL} is equivalent to maximizing the evidence lower bound objective (ELBO).

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\alpha};\mathbf{e})}\left[\log \frac{p(\boldsymbol{\alpha}, \mathcal{Y}_k)}{q(\boldsymbol{\alpha};\mathbf{e})}\right]. \tag{29}$$

The principle is shown in Figure 2. Then, ELBO takes the derivative with respect to parameter **e**:

$$\nabla_{\mathbf{e}}\mathcal{L} = \mathbb{E}_{q(\boldsymbol{\alpha};\mathbf{e})}[f(\boldsymbol{\alpha})], \tag{30}$$

where the following is the case.

$$f(\boldsymbol{\alpha}) = \nabla_{\mathbf{e}} \log q(\boldsymbol{\alpha};\mathbf{e})(\log p(\boldsymbol{\alpha}, \mathcal{Y}_k) - \log q(\boldsymbol{\alpha};\mathbf{e})). \tag{31}$$

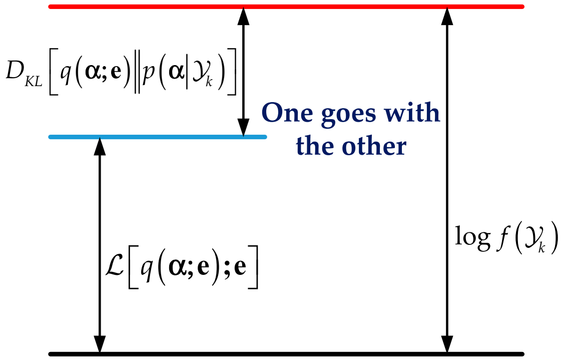

$D_{KL}\left[q(\boldsymbol{\alpha};\mathbf{e})\middle\| p(\boldsymbol{\alpha}|\mathcal{Y}_k)\right]$
One goes with the other
$\log f(\mathcal{Y}_k)$
$\mathcal{L}[q(\boldsymbol{\alpha};\mathbf{e});\mathbf{e}]$

Figure 2. The principle of conversion.

As a result, computing $\nabla_{\mathbf{e}}\mathcal{L}$ becomes a process of looking for an expectation, which can be estimated by the Monte Carlo approach. Then, the gradient descent method is used to make the parameters converge, which makes $q(\boldsymbol{\alpha};\mathbf{e})$ approximate the real posterior distribution.

The aforementioned black-box variational inference (BBVI) only relies on samples from $q(\boldsymbol{\alpha};\mathbf{e})$ to calculate the Monte Carlo gradient. However, the samples of the variational distribution $q(\boldsymbol{\alpha};\mathbf{e})$ are concentrated near the peak point, and there are fewer samples in the tail. This causes the Monte Carlo gradient to often have high variance, which means that the estimated gradient may be very different from the true value, resulting in optimization times that are too long.

In view of this defect, O-BBVI adds a good proposal distribution that matches the variational problem [33]. $r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$ is an overdispersed version of $q(\boldsymbol{\alpha};\mathbf{e})$. Its tail is heavier, which increases the probability that the value of the tail is sampled. In addition, O-BBVI also uses two strategies, control variates and Rao–Blackwellization in order to reduce the variance of the Monte Carlo gradient of the original BBVI. Finally, O-BBVI constructed a new Monte Carlo gradient equation, and based on the unconventional usage of importance sampling, it samples from $r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$ and $q(\boldsymbol{\alpha};\mathbf{e})$, respectively, in order to estimate the gradient.

Let us begin with this overdispersed distribution:

$$r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau}) = g(\boldsymbol{\alpha},\boldsymbol{\tau}) \exp\left\{\frac{\mathbf{e}^T t(\boldsymbol{\alpha}) - A(\mathbf{e})}{\boldsymbol{\tau}}\right\}, \tag{32}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2]$ is the dispersion coefficient of the overdispersed distribution. For a fixed $\boldsymbol{\tau}$, $r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$ and $q(\boldsymbol{\alpha};\mathbf{e})$ belong to the same exponential family. Moreover,$r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$ allocates higher mass to the tails of $q(\boldsymbol{\alpha};\mathbf{e})$, which is why we introduced it.

O-BBVI believes that for each component of the gradient, the proposal distribution that minimizes the variance of the estimator is not $q(\boldsymbol{\alpha};\mathbf{e})$ but the following.

$$q_n^{op}(\boldsymbol{\alpha}) \propto q(\boldsymbol{\alpha};\mathbf{e})|f_n(\boldsymbol{\alpha})|, \tag{33}$$

That is, the optimal proposal distribution pushes the probability density to the tail of $q(\boldsymbol{\alpha};\mathbf{e})$;hence, we have reason to believe that $r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$ is closer to the optimal proposal distribution than the variational distribution $q(\boldsymbol{\alpha};\mathbf{e})$ due to the fact that, in sampling, it provides more opportunities for $\boldsymbol{\alpha}$, which exists in the tail of variational distribution but has higher posterior probability. Thus, (30) can be rewritten as follows:

$$\hat{\nabla}_{\mathbf{e}}\mathcal{L} = \frac{1}{M}\sum_m f\left(\boldsymbol{\alpha}^{(m)}\right)\frac{q\left(\boldsymbol{\alpha}^{(m)};\mathbf{e}\right)}{r\left(\boldsymbol{\alpha}^{(m)};\mathbf{e},\tau\right)}, \tag{34}$$

where $\boldsymbol{\alpha}^{(m)}$ represents the $m-th$ sample in the sample set of $r(\boldsymbol{\alpha};\mathbf{e},\boldsymbol{\tau})$.

To solve the problem of importance sampling failure caused by high-dimensional hidden variables, we use the correlation theory of mean-field, i.e., keep the rest of the components unchanged when solving the $n-th$ component. That is to say, (34) can be expressed as follows:

$$\hat{\nabla}_{e_n}\mathcal{L} = \mathbb{E}_{r(\alpha_n;e_n,\tau_n)}\left[w(\alpha_n)\mathbb{E}_{q(\boldsymbol{\alpha}_{-n};\mathbf{e}_{-n})}[f_n(\boldsymbol{\alpha})]\right], \tag{35}$$

where $w(\alpha_n) = q(\alpha_n;e_n)/r(\alpha_n;e_n,\tau_n)$, and $\boldsymbol{\alpha}_{-n}$ means variables other than α_n. $\mathbf{e}_{-n}$ is the same. Inspired by the foregoing discussion, we take a simple sample $\boldsymbol{\alpha}_{-n}^0$ from $q(\boldsymbol{\alpha}_{-n};\mathbf{e}_{-n})$ to estimate $\mathbb{E}_{q(\boldsymbol{\alpha}_{-n};\mathbf{e}_{-n})}[\cdot]$ and extract M samples from $r(\alpha_n;e_n,\tau_n)$ to estimate $\mathbb{E}_{r(\alpha_n;e_n,\tau_n)}[\cdot]$. At the same time, a score function is defined as follows.

$$h_n\left(\alpha_n^{(m)}\right) = \nabla_{e_n}\log q\left(\alpha_n^{(m)};e_n\right). \tag{36}$$

It is used to calculate the following.

$$f_n\left(\boldsymbol{\alpha}^{(m)}\right) = h_n\left(\alpha_n^{(m)}\right)\log\frac{p_n\left(\mathbf{y}_k,\alpha_n^{(m)},\boldsymbol{\alpha}_{-n}^{(0)}\right)}{q\left(\alpha_n^{(m)};e_n\right)}. \tag{37}$$

This formula is the embodiment of Rao–Blackwellization. Note that in order to reduce the complexity of the algorithm, we utilize $\mathbf{y}_k$ instead of $\mathcal{Y}_k$ in Equation (37). Moreover, p_n is acquired by the Markov assumption. Herein, we assume that $\mathbf{x}_k$ is a Gaussian distribution with $\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$ as the mean and $\mathbf{P}_k^{\mathbf{x},\alpha}$ as the covariance. Given the state $\mathbf{x}_k$, the corresponding observation $\mathbf{y}_k$ obeys the Gaussian distribution $\mathcal{N}(\mathbf{y}_k;\mathbf{H}_k\mathbf{x}_k,\mathbf{R}^{\alpha_2})$. Thus, the likelihood function of the unknown parameter $\boldsymbol{\alpha}$ can be approximated as follows.

$$f(\mathbf{y}_k|\boldsymbol{\alpha}) = \mathcal{N}(\mathbf{y}_k;\mathbf{H}_k\mathbf{x}_k,\mathbf{R}^{\alpha_2}). \tag{38}$$

Consequently, we obtain the following.

$$p_n\left(\mathbf{y}_k,\alpha_n,\boldsymbol{\alpha}_{-n}^{(0)}\right) = f(\mathbf{y}_k|\boldsymbol{\alpha})p_n(\alpha_n)p_{-n}\left(\boldsymbol{\alpha}_{-n}^{(0)}\right). \tag{39}$$

Finally, the gradient of ELBO relative to each component e_n is as follows:

$$\hat{\nabla}_{e_n}\mathcal{L} = \frac{1}{M}\sum_m\left[f_n^w\left(\boldsymbol{\alpha}_n^{(m)}\right) - b_n h_n^w\left(\alpha_n^{(m)}\right)\right] \tag{40}$$

where the following is the case.

$$f_n^w\left(\boldsymbol{\alpha}^{(m)}\right) = w\left(\alpha_n^{(m)}\right) f_n\left(\boldsymbol{\alpha}^{(m)}\right), \tag{41}$$

$$h_n^w\left(\alpha_n^{(m)}\right) = w\left(\alpha_n^{(m)}\right) h_n\left(\alpha_n^{(m)}\right), \tag{42}$$

$$b_n = \frac{Cov(f_n^w, h_n^w)}{var(h_n^w)}. \tag{43}$$

Note that b_n marks the application of the strategy of control variates.

Next, we use the AdaGrad algorithm to make $\mathbf{e}$ converge to the optimal value:

$$\mathbf{e}^{(t)} = \mathbf{e}^{(t-1)} + \boldsymbol{\lambda}_t \circ \hat{\nabla}_{\mathbf{e}}\mathcal{L}, \tag{44}$$

where $\boldsymbol{\lambda}_t$ is the learning rate, and its setting is similar to that in [34]. $'\circ'$ stands for the Hadamard product. Note that the expectations contained in natural parameter $\mathbf{e}$ are the posterior noise statistics $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathbf{y}_k]$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathbf{y}_k]$ of the k-th sampling period and not the $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]$ and $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k]$ we require. Here, they are only needed for computing the average value from 0 to the k-th sampling period.

Furthermore, we want to automatically adjust the dispersion coefficient τ_n during the iteration. We first calculate the derivative of (35) with respect to τ_n.

$$-\frac{\partial \mathbb{V}[\hat{\nabla}_{e_n}\mathcal{L}]}{\partial \tau_n} = \frac{1}{M}\mathbb{E}_{r(\alpha_n;e_n,\tau_n)}\left[\mathbb{E}_{q(\boldsymbol{\alpha}_{-n};\mathbf{e}_{-n})}[f_n(\boldsymbol{\alpha})]^2 w^2(\alpha_n)\frac{\partial \log r(\alpha_n;e_n,\tau_n)}{\partial \tau_n}\right]. \tag{45}$$

Since $f_n(\boldsymbol{\alpha})$ and $w(\alpha_n)$ are calculated in the above process, this process will not consume much time. Next, the Monte Carlo estimation method is used to calculate the solution of the equation and then the gradient descent method is used to make t converge:

$$\tau_n^{(t)} = \tau_n^{(t-1)} - s_n\frac{\partial \mathbb{V}[\hat{\nabla}_{e_n}\mathcal{L}]}{\partial \tau_n}, \tag{46}$$

where s_n is the step size, which needs to be set in advance. Note that if e_n represents a vector, only the derivative part needs to be adjusted to the sum of the derivatives of all components of e_n, and Equation (46) can still be used.

Finally, if the posterior noise statistics converge to a certain value or changes little after multiple iterations, the subsequent state estimation can skip the step of finding the posterior noise parameters in order to save the algorithm overhead. The steps for implementation of the proposed KFPNS framework are summarized in Algorithms 1 and 2.

Algorithm 1: KFPNS.

```
input: $\mathcal{Y}_k, p(\boldsymbol{\alpha}), \boldsymbol{\Phi}_k, \boldsymbol{\Gamma}_k, \mathbf{H}_k$, mean − field variational family $q(\boldsymbol{\alpha};\mathbf{e})$
output: $\hat{\mathbf{x}}_k^{\alpha}$
Initialize $\mathbf{e}, \boldsymbol{\tau}$
  $\hat{\mathbf{x}}_0^{\alpha} \leftarrow \mathbb{E}[\mathbf{x}_0]$
  $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_0^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{-1}\right] \leftarrow$ initialization
  $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_{-1}], \mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_{-1}] \leftarrow$ Empirical value
  $k \leftarrow 0$
for $k = 1, 2, \ldots$ do
    $\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}} \leftarrow \mathbf{y}_k^{\boldsymbol{\alpha}} - \mathbf{H}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}}$
    $\mathbf{K}_k^{\Theta'} \leftarrow \mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{k-1}\right]\mathbf{H}_k^T\mathbb{E}_{\boldsymbol{\alpha}}{}^{-1}\left[\mathbf{H}_k\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}\mathbf{H}_k^T + \mathbf{R}^{\alpha_2}|\mathcal{Y}_{k-1}\right]$
    $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k], \mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k] \leftarrow O - \mathrm{BBVI}(\mathbf{y}_k, p(\boldsymbol{\alpha}))$
    $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_{k+1}^{\mathbf{x},\boldsymbol{\alpha}}\middle|\mathcal{Y}_k\right] \leftarrow \boldsymbol{\Phi}_k\left(\mathbf{I} - \mathbf{K}_k^{\Theta'}\mathbf{H}_k\right)\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_k^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_k\right]\boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]\boldsymbol{\Gamma}_k^T$
    $\hat{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\Phi}_k\hat{\mathbf{x}}_k^{\boldsymbol{\alpha}} + \boldsymbol{\Phi}_k\mathbf{K}_k^{\Theta}\widetilde{\mathbf{z}}_k^{\boldsymbol{\alpha}}$
      return $\hat{\mathbf{x}}_{k+1}^{\boldsymbol{\alpha}}$
end for
```

Algorithm 2: O-BBVI.

```
function O-BBVI ($\mathbf{y}_k, p(\boldsymbol{\alpha})$)
$f(\mathbf{y}_k|\boldsymbol{\alpha}) \leftarrow \mathcal{N}(\mathbf{y}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}^{\alpha_2})$
While the algorithm has not converged do
    Draw $\boldsymbol{\alpha}^{(0)} \sim q(\boldsymbol{\alpha}; \mathbf{e})$
    for $n = 1$ to 2 do
        Draw $M$ samples $\alpha_n^{(m)} \sim r(\alpha_n; e_n, \tau_n)$
        $p_n\left(\mathbf{y}_k, \alpha_n, \boldsymbol{\alpha}_{-n}^{(0)}\right) \leftarrow f(\mathbf{y}_k|\boldsymbol{\alpha})p_n(\alpha_n)p_{-n}\left(\boldsymbol{\alpha}_{-n}^{(0)}\right)$
        for $m = 1$ to $M$ do
            $w\left(\alpha_n^{(m)}\right) = q\left(\alpha_n^{(m)}; e_n\right)/r\left(\alpha_n^{(m)}; e_n, \tau_n\right)$
            $h_n\left(\alpha_n^{(m)}\right) \leftarrow \nabla_{e_n} \log q\left(\alpha_n^{(m)}; e_n\right)$
            $f_n\left(\boldsymbol{\alpha}^{(m)}\right) \leftarrow h_n\left(\alpha_n^{(m)}\right) \log\left(p_n\left(\mathbf{y}_k, \alpha_n^{(m)}, \boldsymbol{\alpha}_{-n}^{(0)}\right)/q\left(\alpha_n^{(m)}; e_n\right)\right)$
            $f_n^w\left(\boldsymbol{\alpha}^{(m)}\right) \leftarrow w\left(\alpha_n^{(m)}\right)f_n\left(\boldsymbol{\alpha}^{(m)}\right)$
            $h_n^w\left(\alpha_n^{(m)}\right) \leftarrow w\left(\alpha_n^{(m)}\right)h_n\left(\alpha_n^{(m)}\right)$
        end for
        $b_n \leftarrow Cov(f_n^w, h_n^w)/Var(h_n^w)$
        $\hat{\nabla}_{e_n}\mathcal{L} \leftarrow \sum_m \left[f_n^w\left(\boldsymbol{\alpha}_n^{(m)}\right) - b_n h_n^w\left(\alpha_n^{(m)}\right)\right]/M$
    end for
    for $n = 1$ to 2 do
        $-\partial\mathbb{V}[\hat{\nabla}_{e_n}\mathcal{L}]/\partial\tau_n \leftarrow$ Eq.45
        $\tau_n \leftarrow$ Eq.46
    end for
    set up $\boldsymbol{\lambda}_t$
    $\mathbf{e}_k^{(t)} \leftarrow \mathbf{e}_k^{(t-1)} + \boldsymbol{\lambda}_t \circ \hat{\nabla}_{\mathbf{e}}\mathcal{L}$
$\overset{\frown}{\mathbf{e}} \leftarrow \text{mean}[\mathbf{e}_0, \cdots, \mathbf{e}_k]$
$\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k] \leftarrow \overset{\frown}{e}_1$, $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k] \leftarrow \overset{\frown}{e}_2$
return $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{Q}^{\alpha_1}|\mathcal{Y}_k]$, $\mathbb{E}_{\boldsymbol{\alpha}}[\mathbf{R}^{\alpha_2}|\mathcal{Y}_k]$
```

3. Results

In this section, we analyze the performance of the proposed algorithm and other Kalman filtering methods. The first Kalman filtering method to participate in the comparison is the IBR approach. The second one is the model-specific Kalman filtering approach, which is a classical Kalman filter designed with respect to real noise parameters. The third is the minimax method, which performs best in the worst case when the noise parameters are uncertain. We set the parameters of the minimax method to $\alpha_{\max 1} = 4$, $\alpha_{\max 2} = 4$. The last one is OBKF, which designs a message passing algorithm based on factor graphs to calculate the likelihood function formulaically and then employs the Metropolis Hastings MCMC method to find posterior effective noise statistics.

3.1. Simulation

For the tracking scenario in a two-dimensional space, we assume that the state of the target is $\mathbf{x}_k = \left[p_x p_y v_x v_y\right]^T$, and the subscripts x, y represent the x and y dimensions, respectively. Assuming that the target's motion model is a constant velocity model and that there is only a weak random disturbance, the radar obtains the measurements of the target at every t seconds. We construct the state-space model of the measured target based on (9) and (10), where the matrices' representations are as follows.

$$\boldsymbol{\Phi}_k = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \boldsymbol{\Gamma}_k = \begin{bmatrix} t^2/2 & 0 \\ 0 & t^2/2 \\ t & 0 \\ 0 & t \end{bmatrix}.$$

The covariance matrices of the process and observation noise are expressed as follows.

$$\mathbf{Q}^{\alpha_1} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_1 \end{bmatrix}, \mathbf{R}^{\alpha_2} = \begin{bmatrix} \alpha_2 & 0 \\ 0 & \alpha_2 \end{bmatrix}.$$

Let $t = 1$ and the target's motion state is initialized to $\mathbb{E}[\mathbf{x}_0] = [1035 - 5]^T$ and $\mathbb{E}_{\boldsymbol{\alpha}}\left[\mathbf{P}_0^{\mathbf{x},\boldsymbol{\alpha}}|\mathcal{Y}_{-1}\right] = diag([1000\ 1000\ 100\ 100\ ])$, where $diag(\mathbf{a})$ represents a diagonal matrix with elements in vector **a** as diagonal elements.

In the first simulation, it is assumed that both α_1 and α_2 are unknown parameters that are uniformly distributed in the interval $[1,4]$ and $[0.5,4]$, respectively. In order to analyze the average performance of five Kalman filtering methods, we generate a large amount of data based on the model described above and calculate the average MSE. The results are shown in Figure 3. This set of data is generated depending on the prior distribution of noise, namely, we randomly generated 200 pairs $[\alpha_1, \alpha_2]$, and each combination corresponds to 10 different observation sequences. Among the five filters, the average MSE of the model-specific Kalman filter is the smallest because it uses real noise parameters. Instead, the minimax approach only considers the worst case and cannot effectively solve the problem of unknown noise second-order statistics. Figure 3 also suggests that the average MSEs of OBKF and IBRKF are almost the same at the initial stage of filtering, and both greatly outperform KFPNS since the performance of KFPNS depends on the length of the observation sequence used to adjust filter parameters and the prior distribution of noise. If the number of sampling period k is small, the filter parameters may be adjusted incorrectly after each iteration, resulting in large state estimation errors for certain subsequent sampling periods. However, in the long run, the increase in analyzable data will return the uncertain parameters close to the true value. At this time, the performance of the KFPNS is even similar to that of OBKF. Both KFPNS and OBKF need to use a certain amount of observed data to estimate unknown noise parameters in order to achieve better estimation results, which causes them to converge more slowly than other algorithms, especially for KFPNS. The good news is that as the number of observations continues to increase, their performance is closer to the model-specific Kalman filter.

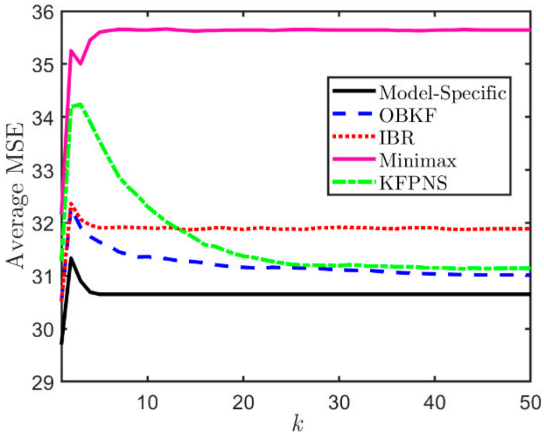


Figure 3. Average MSE of various filters when α_1 and α_2 are uncertain.

We set up two pairs of specific noise parameters for the second set of simulations. In order to discuss the average performance of various filters, we provide 200 observation sequences for each specific combination in order to weaken the interference of abnormal results. Then, we calculated the average MSE of 200 sets of observations in order to visually compare the performance of various filters. The first column of Figure 4 is a performance analysis based on the specific model $\alpha_1 = 3, \alpha_2 = 1.5$. The true value of the specific noise pair in the second column is $\alpha_1 = 3.5, \alpha_2 = 3.5$. The variation of average MSE of each filter in Figure 4a is roughly consistent with the previous simulation. Figure 4c,e, respectively, show the variation of the average posterior mean $\mathbb{E}[\alpha_1|\mathcal{Y}_k]$ and $\mathbb{E}[\alpha_2|\mathcal{Y}_k]$ under the first specific model. Similarly, (d) and (f) correspond to the second specific noise combination. The vertical columnar line represents the variance, and the size of variance is represented by the length of the line. The figures in the second and third rows show that with the increase in observation data, $\mathbb{E}[\alpha_1|\mathcal{Y}_k]$ and $\mathbb{E}[\alpha_2|\mathcal{Y}_k]$, moves closer to the true value from the preset empirical values of 2.2 and 2 and finally stabilizes near the true value. At the same time, the variance of the average posterior also means changes from small to large and then to small and eventually stabilizes. This indicates that the posterior distribution of noise estimated by KFPNS is more and more in line with the true distribution, and its value is concentrated around the true value. We also observe that in Figure 4b, the minimax method outperforms IBRKF and even outperforms OBKF in the beginning. Note that under the condition of prior knowledge, the average performance of IBRKF exceeds the minimax Kalman filter. However, IBRKF performs worse with respect to some specific models; for example, when the parameters of the minimax method are closer to the true values than the prior mean of IBRKF. Fortunately, both the OBKF and KFPNS use their approaches to approximate true posterior noise distribution, and as we observe more data, posterior second-order statistics will tend towards the true model. This allows them to break the limitations of most noise models and to deal with models that IBRKF is unable to handle.

Next, we discuss the performance changes of different robust Kalman filters when the mass of the prior probability is not uniform. Suppose α_1 is uniformly distributed in the interval $[1,2]$ and fixed. On the other hand, let a Beta distribution $\mathcal{B}(\alpha_r, \beta_r)$ in the interval $[0.25, 4]$ govern α_2. The mean and variance of the Beta distribution are $\alpha_r/\alpha_r + \beta_r$ and $\alpha_r\beta_r/(\alpha_r + \beta_r)^2(\alpha_r + \beta_r + 1)$, respectively, and $\alpha_r + \beta_r = 1$. On the basis of ensuring that the mean of this prior distribution remains unchanged, a new prior distribution with different mass can be obtained by appropriately adjusting its parameters. We consider two pairs of specific parameters, $\alpha_r = 0.1, \beta_r = 0.9$ and $\alpha_r = 0.01, \beta_r = 0.09$. As a result, the reduction in parameters causes the variance to become larger, and the prior distribution will be more relaxed. Figure 5 shows their average MSEs. Even if the Beta distribution is changed to make the probability of values in prior distribution different, the average MSEs of the minimax method and IBRKF are almost unchanged. The reason is that for all prior distributions, the prior mean of the IBR method is always the same, while the minimax method does not consider prior distribution at all. They cannot effectively cope with sudden changes in the model. In this case, they are equivalent to the classical Kalman filter. Compared with the above two robust filtering strategies, although the mass distribution of the prior has changed, since KFPNS incorporates observations into prior knowledge in order to estimate posterior noise distribution, its average MSE is related to the tightness of the prior distribution. In other words, the tighter the prior distribution, the worse the performance of KFPNS, while the opposite is true when the prior distribution is relaxed. Importantly, the average MSE of KFPNS can always be closer to that of the model-specific Kalman filter.

In the final simulation, we analyze the performance of each robust filtering strategy when the prior knowledge is inaccurate. In practical applications, it is often impossible to understand the underlying true model, and the mastery of prior knowledge is not comprehensive. Therefore, it is crucial for a robust filter not to rely too much on the prior distribution. Our previous simulations are based on the assumption that the prior distribution of noise is accurate and available. Here, we only consider the case where the variance of the observation noise is unknown and the range of its prior distribution is included in the interval of the true distribution. We still have α_1 uniformly distributed over $[1,2]$, while assuming that α_2 is uniformly distributed over $[3,5]$, which is wrong relative to the real interval. Similarly, we generated 20 different true values based on the correct prior interval, and each true value contained 20 different sets of observations. The evaluation index of the robustness of each filter is still the average MSE. Figure 6a,b are the average MSEs of various Kalman filters in the entire sampling period when the correct interval is $[2,6]$ and $[0.5,7.5]$. When k is large, the two robust filters, OBKF and KFPNS still perform best in terms of average MSEs. It is worth noting that KFPNS outperforms OBKF for the first time in Figure 6b, but it is slightly inferior or equivalent to OBKF in the previous simulation. This is related to the computing method of posterior noise statistics. When OBKF employs the Metropolis Hastings MCMC method to select sample points, the prior distribution incorporated is wrong and lacks an adjustment strategy, which makes the posterior noise statistics unable to converge to the true value. On the contrary, KFPNS adds a weighting factor in the calculation process, making it not completely dependent on the inaccurate prior distribution. Therefore, KFPNS performs better when facing models with larger errors between prior knowledge and the underlying true model. Unfortunately, the performance of KFPNS may decline when the prior distribution provided is inaccurate, and the stronger the inaccuracy, the faster the performance decline.

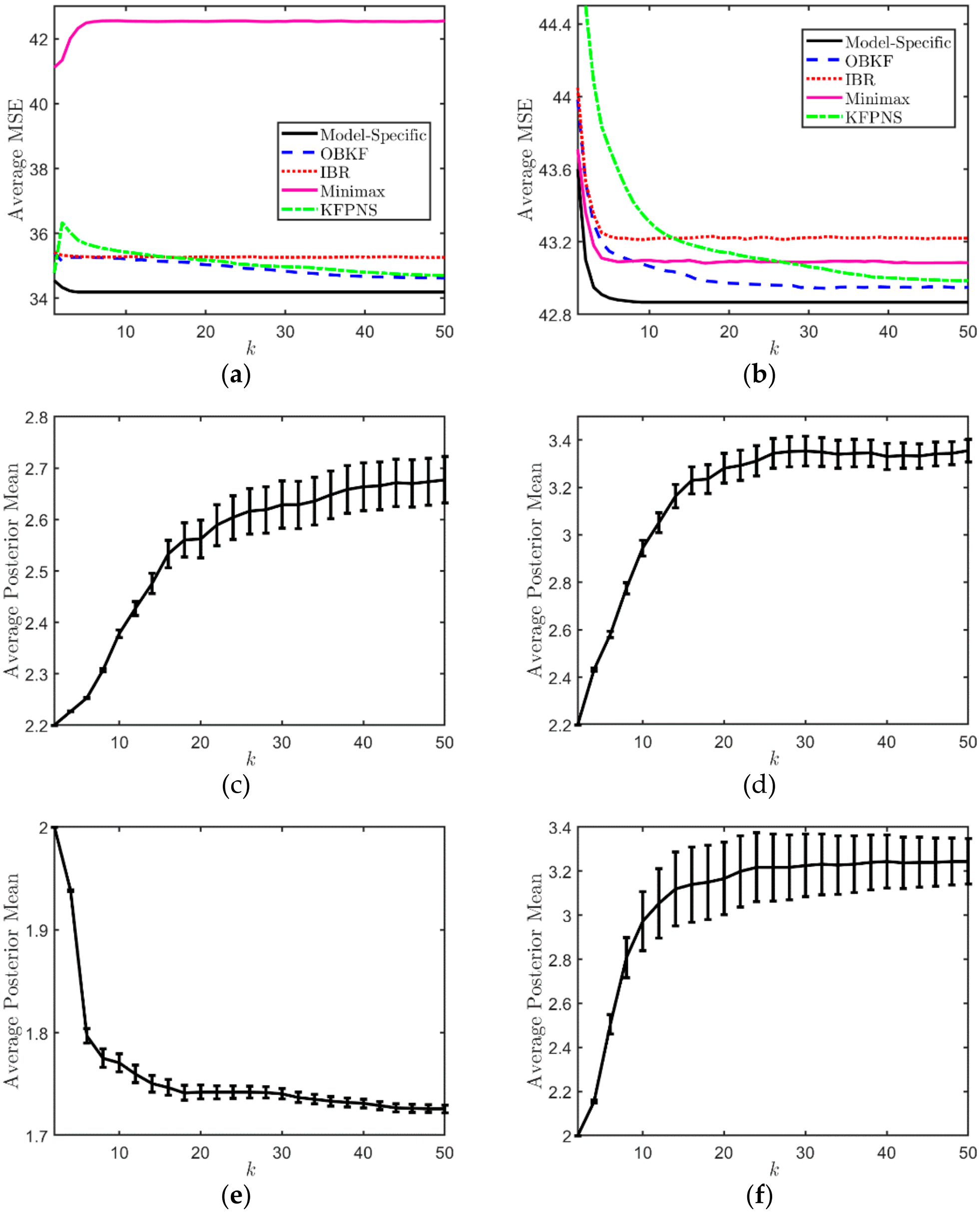


Figure 4. Performance analysis for specific noise pairs. (**a**) The average MSE for specific noise model $\alpha_1 = 3, \alpha_2 = 1.5$. (**b**) The average MSE for specific noise model $\alpha_1 = 3.5, \alpha_2 = 3.5$. (**c**) The variation of $\mathbb{E}[\alpha_1|\mathcal{Y}_k]$ and its variance when $\alpha_1 = 3$. (**d**) The variation of $\mathbb{E}[\alpha_1|\mathcal{Y}_k]$ and its variance when $\alpha_1 = 3.5$. (**e**) The variation of $\mathbb{E}[\alpha_2|\mathcal{Y}_k]$ and its variance when $\alpha_2 = 1.5$. (**f**) The variation of $\mathbb{E}[\alpha_2|\mathcal{Y}_k]$ and its variance when $\alpha_2 = 3.5$.

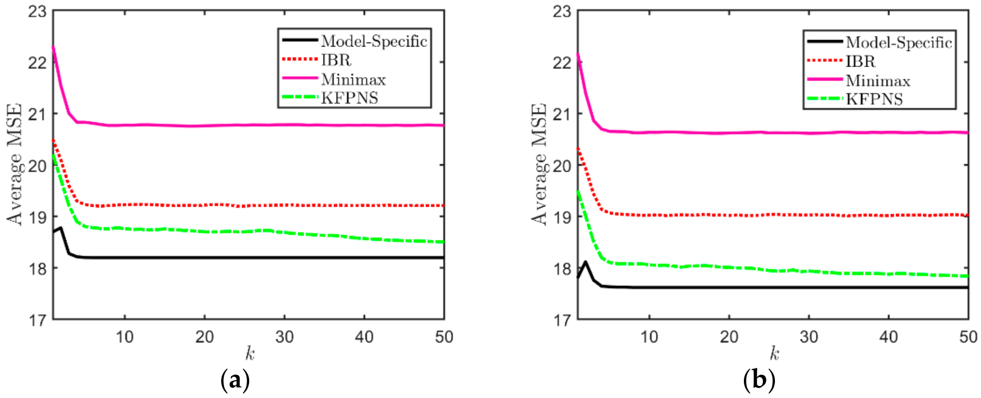


Figure 5. The average performance of four filters with different Beta priors. (**a**) $\alpha_r = 0.1, \beta_r = 0.9$; (**b**) $\alpha_r = 0.01, \beta_r = 0.09$.

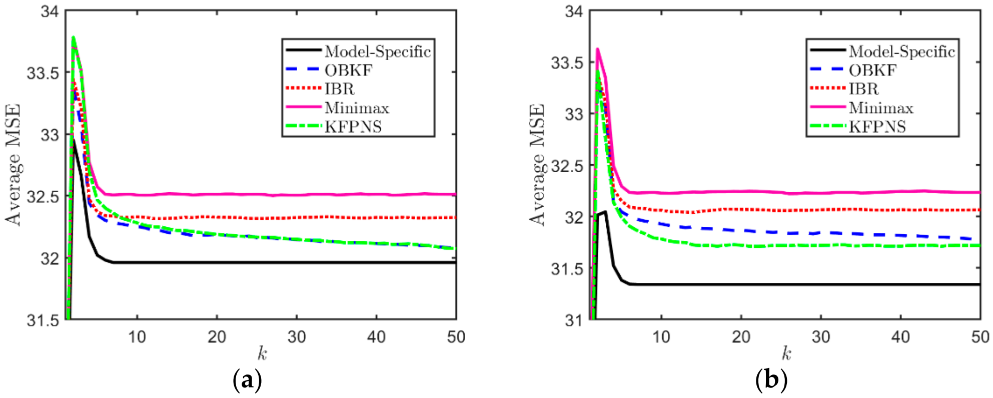


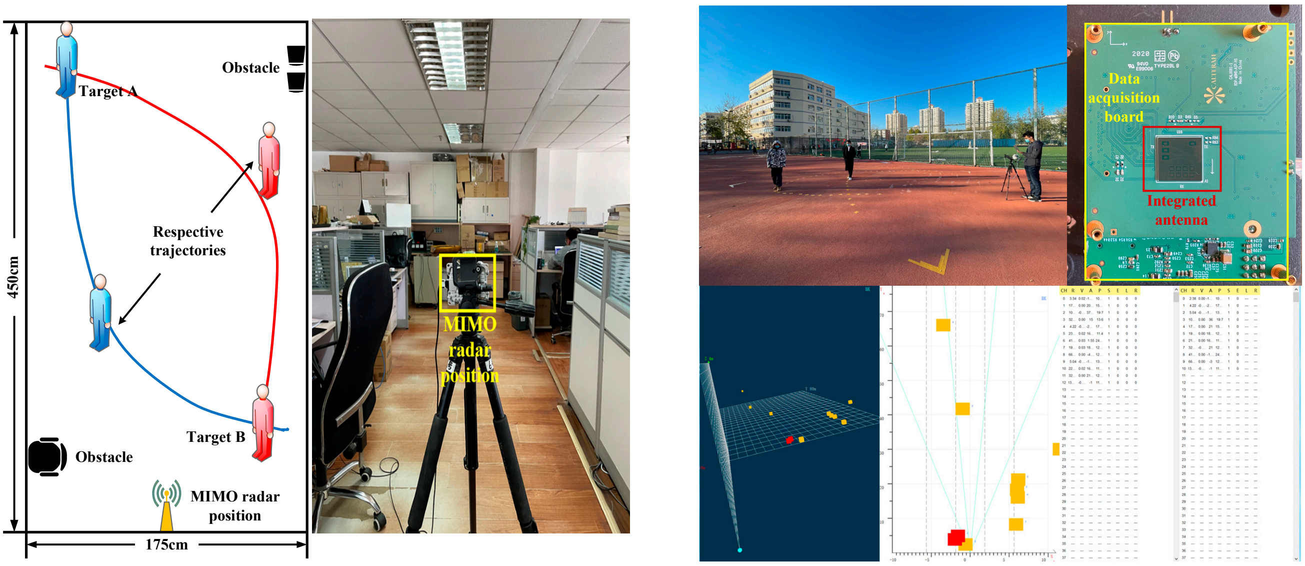
Figure 6. Average MSE of various filters when α_2 is uncertain, and the interval $[3, 5]$ of prior distribution is inaccurate. (**a**) The exact interval is $[2, 6]$. (**b**) The exact interval is $[0.5, 7.5]$.

3.2. Experiment

We also processed the observed data of the real MIMO radar system. We conducted experiments in two specific scenarios. The experimental conditions are described as follows. We have delineated two fixed experimental areas, namely the indoor area of 175 × 465 cm^2 and the outdoor area of 800 × 800 cm^2. In addition, we also calibrated some points in the two experimental areas and took these points as the ground truth of the target. Due to the limited scanning range of the MIMO radar, the position of the radar has a certain distance from the boundary of the experimental area, but we eliminated the influence of this distance in the subsequent processing. Table 2 reports the parameters of the radar used. We first sampled the environment of the two experimental areas without any targets and filtered out non-target points on this basis. During data acquisition, two people moved at a constant velocity in these two areas according to two predetermined trajectories. Figure 7 describes our experimental scenarios. Figure 7b also shows the MIMO radar board we used and the host computer interface during data acquisition. After collecting measurements, we clustered the target points and took the cluster center as the observed position of the target. Finally, four different Kalman filtering methods were used to process the data and to analyze the error.

Table 2. MIMO radar parameters.

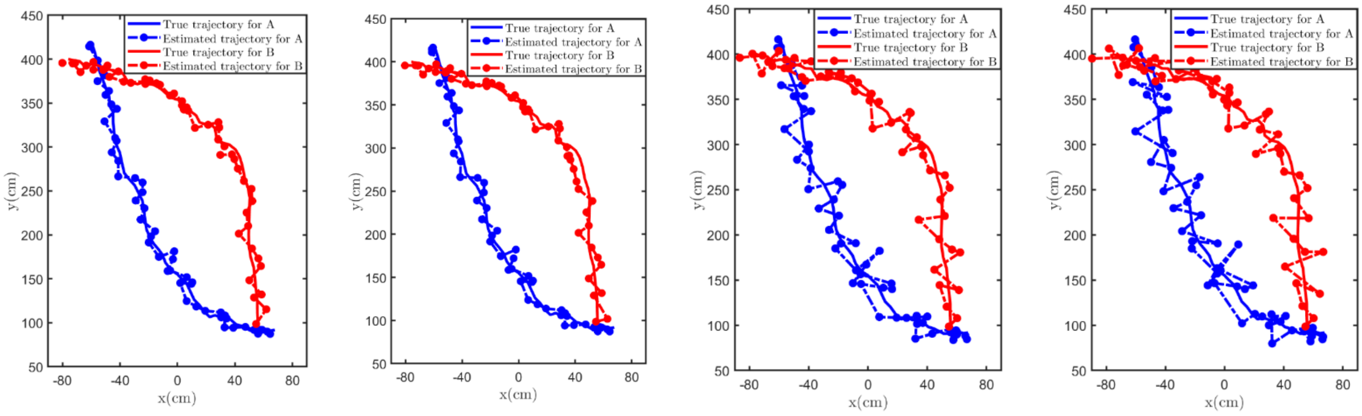
Radar Model	Radar System	The Start Frequency	Range Resolution	Frame Periodicity	Scan Range
RDP-77S244-ABM-AIP	FMCW	77 GHz	0.045 m	200 ms	FOV 120°



(**a**) (**b**)

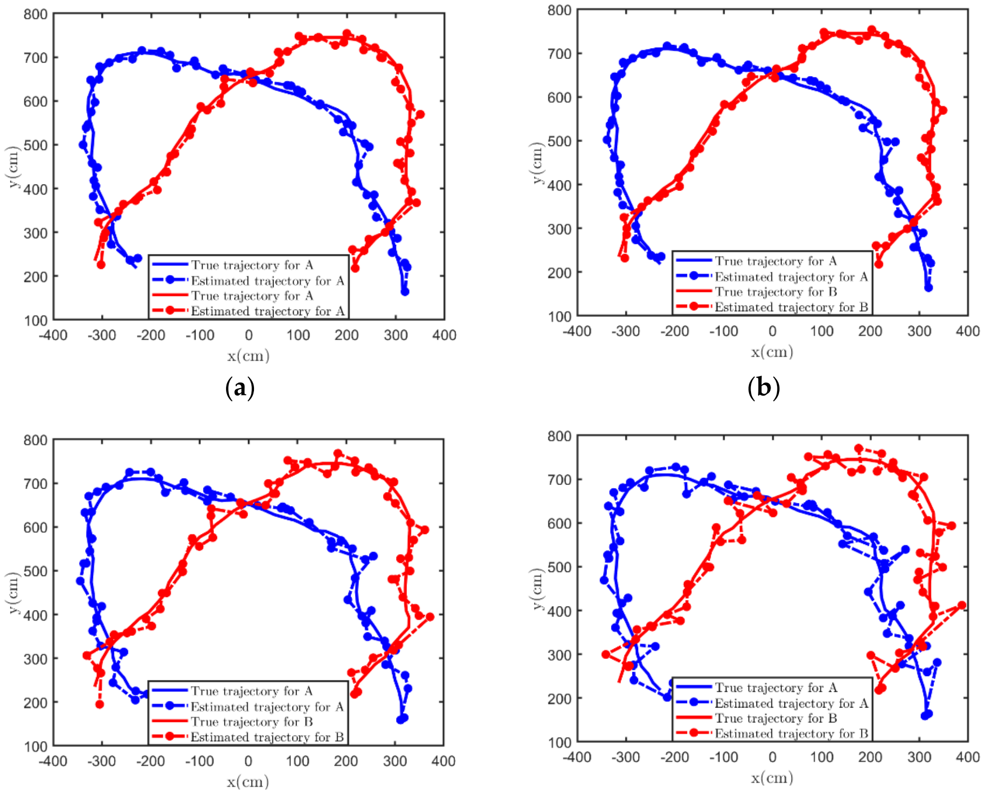
Figure 7. Real data acquisition scenes. (**a**) Indoor scene. (**b**) Outdoor scene.

The filters we use are the classical Kalman filter (CKF), IBRKF, OBKF and KFPNS. The filtering results under different experimental scenarios are shown in Figures 8 and 9. Note that the blue point represents the estimated position of target A, and the red point represents the estimated position of target B. When filtering these position data, we do not fully grasp the underlying real model, which means that the prior knowledge we provided for these algorithms may not be accurate enough. Nevertheless, it can be observed that the estimated trajectories of the two Kalman filters designed based on posterior information are significantly smoother than the other methods. The inaccuracy of prior knowledge of noise is the main reason for the poor trajectory estimation of CKF and IBRKF. Although IBRKF introduces the concept of effective statistics, which weakens this effect to a certain extent, when the inaccuracy of prior knowledge is strong, its performance is only slightly improved compared with the classical algorithm. As expected, both KFPNS and OBKF have the characteristics of computing the posterior value and making it approximate the real value through algorithm iteration, which makes them very robust; thus, their estimated trajectory is also the closest to the true trajectory.



(**a**) (**b**) (**c**) (**d**)

Figure 8. The fitting degree between the estimated trajectory and the true trajectory after processing by various Kalman filtering methods in an indoor scene. (**a**) OBKF. (**b**) KFPNS. (**c**) IBRKF. (**d**) CKF.



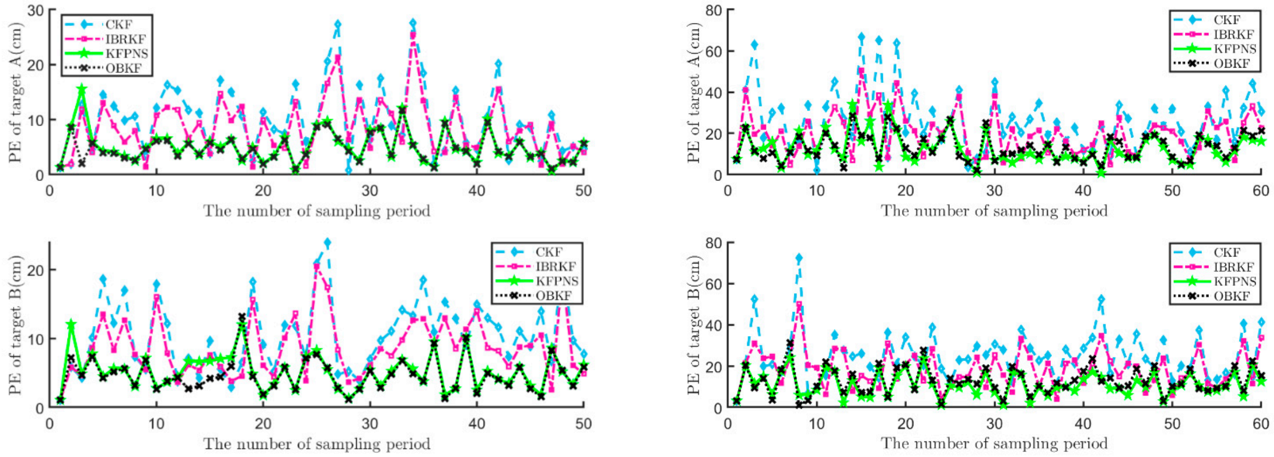
(**a**) (**b**)

(**c**) (**d**)

Figure 9. The fitting degree between the estimated trajectory and the true trajectory after processing by various Kalman filtering methods in an outdoor scene. (**a**) OBKF. (**b**) KFPNS. (**c**) IBRKF. (**d**) CKF.

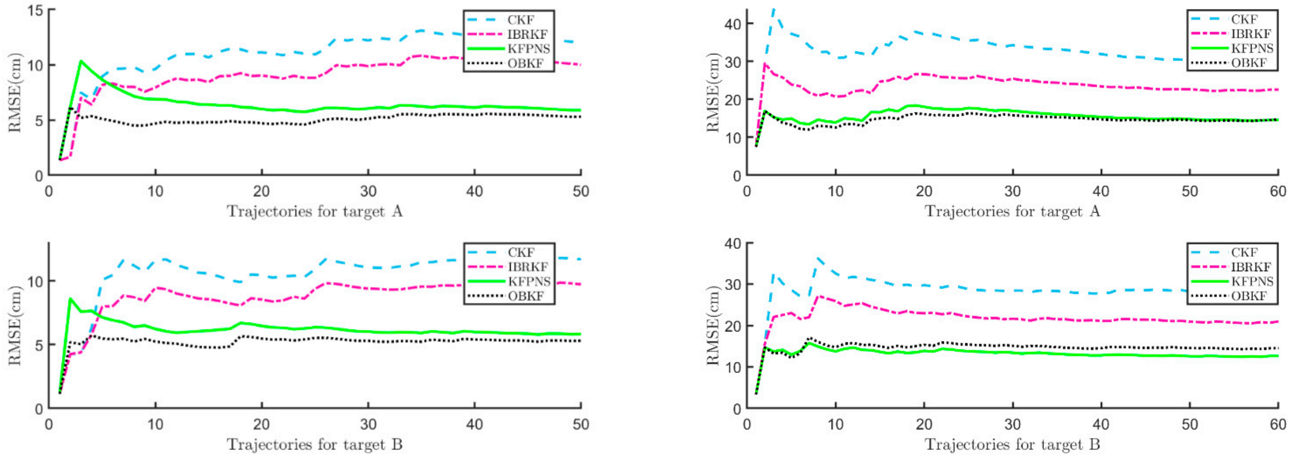
In addition, the position error (PE) between the estimated position and the true position will also be used as an evaluation index for filter performance. In Figure 10, although the PE of the new filter sometimes exceeds IBRKF and the classical Kalman filter, overall, most of the PE curve of KFPNS is in a lower position. Moreover, the PE of the proposed algorithm is more concentrated than compared to these two algorithms. However, in Figure 10a, it cannot be ignored that KFPNS performs poorly in the early stages of filtering. The reason for this phenomenon may be that the selection of the initial value is quite different from the real value, and there are fewer observed data available for analysis. Fortunately, with the input of more observed data, its estimation error gradually approaches OBKF.

We also compute the root-mean-square error (RMSE) of the four algorithms, and the results are shown in Figure 11. In Figure 11a, when the number of sampling period is small, the RMSE of KFPNS takes the maximum value. After 20 sampling periods, the RMSE of KFPNS tends to be stable and is significantly different from that of the other two Kalman filters with poor robustness. Figure 11b also suggests that the gap between KFPNS and OBKF is further narrowed. The change of RMSE over time proves that both KFPNS and OBKF have strong robustness. Table 3 reports the mean value of RMSE (MMSE) for various algorithms, which can be used to intuitively compare the average performance of various filters in different scenes. In the indoor scene, for target B, the average performance of KFPNS is 29.62% and 41.56% higher than that of IBRKF and the classical Kalman filter separately. Despite the average performance of KFPNS decreases by 18.1% compared with OBKF, they differ by only 0.9434 cm.



(**a**) (**b**)

Figure 10. The PE of each filter in two specific scenes. (**a**) Comparison of PE of various Kalman filters in an indoor scene. (**b**) Comparison of PE of various Kalman filters in an outdoor scene.



(**a**) (**b**)

Figure 11. The RMSE of various Kalman filtering methods in different scenes. (**a**) The RMSE of the four Kalman filtering methods in an indoor scene with respect to target A and target B, respectively. (**b**) The RMSE of the four Kalman filtering methods in an outdoor scene with respect to target A and target B, respectively.

Table 3. The MMSE difference of various Kalman filtering methods.

Algorithm	**MMSE**			
	Indoor Scene		**Outdoor Scene**	
	Target A	**Target B**	**Target A**	**Target B**
KFPNS	6.3679 cm	6.1552 cm	15.5520 cm	13.1918 cm
IBRKF	9.0658 cm	8.7459 cm	23.5724 cm	21.7488 cm
CKF	11.0092 cm	10.5480 cm	32.6647 cm	28.4253 cm
OBKF	5.0358 cm	5.2118 cm	14.6269 cm	14.6148 cm

Figure 12 further analyzes the cumulative distribution function (CDF) of the RMSE of each filtering algorithm in the indoor scene. In Figure 12, 90% of the RMSE of KFPNS is less than 6.739 cm, which improves by 31% over IBRKF (9.767 cm) and 42.35% over CKF (11.69 cm) but degrades by 23.11% over OBKF (5.474 cm). In a nutshell, compared with the IBR robust filtering strategy, KNPNS and OBKF are more suitable for models where the noise is unknown and the real prior cannot be fully grasped.

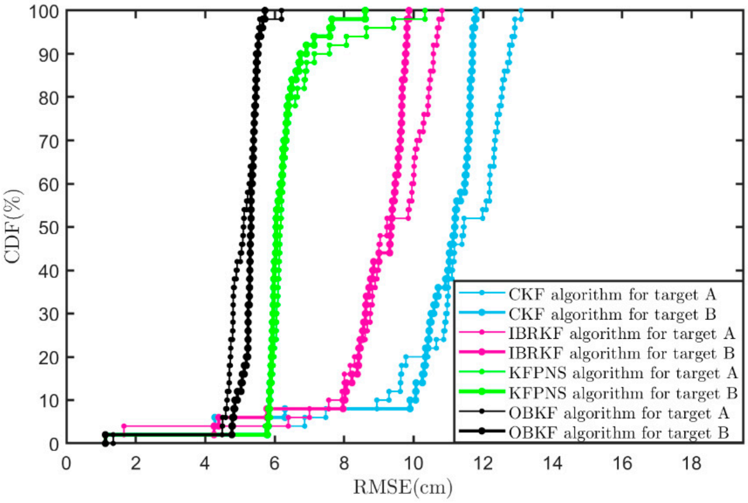


Figure 12. CDFs of RMSE for various algorithms in an indoor scene.

3.3. Time Cost Analysis

Previous experiments have proved that OBKF and KFPNS have considerable robustness. Now we study the difference of their complexity. Since the recursive structure of these two robust Kalman filters is consistent with the classical algorithm, their computational burden is mainly concentrated on the calculation of posterior noise statistics. It is worth mentioning that OBKF uses a factor-graph-based method to convert the problem of finding the likelihood function into a matrix operation, which increases the complexity of the algorithm. Moreover, if the posterior noise statistics at the k-th observation need to be computed, the likelihood function of each MCMC sample must be iterated from $i = 0$ until $i = k - 1$. Therefore, its computational complexity depends on the dimension of the matrix, the number of samples in the posterior distribution and the number of sampling periods. In contrast, the proposed algorithm rarely involves matrix operations, and there is no need to compute the likelihood function from the beginning. At the same time, the variational distribution is fixedly decomposed into two factors. These reasons all reduce the calculation burden of KFPNS to a considerable extent. Figure 13 shows the average run times of the OBKF and KFPNS relative to the sampling period. Note that both algorithms need to generate 10,000 samples to approximate the true posterior noise distribution. In addition, the running platform of all simulations and experiments in this paper is MATLAB R2018b with Intel(R) Core(TM) i5-8257U CPU at 1.40 GHz as well as 8-GB RAM. Figure 13 illustrates that under the same number of samples and the same length of observation sequence, OBKF needs to consume more time. Consequently, KFPNS performs better in real time and is more suitable for applications in actual projects.

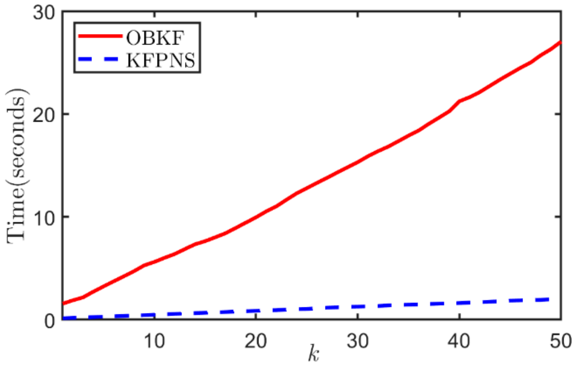


Figure 13. The time consumption of two algorithms when the length of the observation sequence changes.

4. Discussion

In this paper, the IBR framework is extended to a posteriori version by using posterior effective noise information. More exactly, a Bayesian robust Kalman filter based on posterior noise statistics is designed by adding the calculation method of posterior noise statistics. Our simulation results also suggest that when dealing with models with uncertain noise parameters, the performance of the designed filter is second only to OBKF. Therefore, it can be concluded that the proposed calculation method of posterior noise statistics can make the uncertain noise parameters converge near the true value. When used in an actual MIMO radar system, we found that our algorithm has a good performance in robustness and real-time situations compared with current methods. However, it is worth noting that since the run time of the KFPNS is proportional to the length of the observation sequence, when the tracking period is longer, it will consume more time than compared to the classic algorithm. Furthermore, our algorithm is run under the assumption of a Gaussian model. In other words, if the noise is non-Gaussian, the performance of our algorithm will be greatly reduced. In future investigations, we will propose a more effective and faster method for estimating posterior noise distribution and make more classes of noise suitable for this framework.

Author Contributions: Conceptualization, L.C. and C.Z.; data curation, C.F.; formal analysis, Z.Z.; investigation, K.D.; methodology, C.Z.; project administration, L.C.; resources, D.W.; software, Z.Z.; supervision, J.G.; validation, D.W.; writing—original draft, L.C. and C.Z.; writing—review and editing, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Number U20A20163), Scientific Research Project of Beijing Municipal Education Commission (Numbers KZ202111232049 and KM202011232021) and Qin Xin Talents Cultivation Program (Number QXTCP A201902).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kuznetsov, V.P. Stable detection when the signal and spectrum of normal noise are inaccurately known. *Telecommun. Radio Eng.* **1976**, *3031*, 58–64.
2. Kassam, S.A.; Lim, T.L. Robust Wiener filters. *J. Frankl. Inst.* **1977**, *304*, 171–185. [CrossRef]
3. Kalman, R.E. A new approach to linear filtering prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
4. Kia, S.S.; Rounds, S.; Martinez, S. Cooperative Localization for Mobile Agents: A Recursive Decentralized Algorithm Based on Kalman-Filter Decoupling. *IEEE ControlSyst. Mag.* **2016**, *36*, 86–101.
5. Bouzera, N.; Oussalah, M.; Mezhoud, N.; Khireddine, A. Fuzzy extended Kalman filter for dynamic mobile localization in urban area using wireless network. *Appl. Soft Comput.* **2017**, *57*, 452–467. [CrossRef]
6. Safaei, A.; Mahyuddin, M.N. Adaptive Cooperative Localization Using Relative Position Estimation for Networked Systems with Minimum Number of Communication Links. *IEEE Access* **2019**, *7*, 32368–32382. [CrossRef]
7. He, J.J.; Sun, C.K.; Zhang, B.S.; Wang, P. Variational Bayesian-Based Maximum Correntropy Cubature Kalman Filter with Both Adaptivity and Robustness. *IEEE Sens. J.* **2021**, *21*, 1982–1992. [CrossRef]
8. Narasimhappa, M.; Mahindrakar, A.D.; Guizilini, V.C.; Terra, M.H.; Sabat, S.L. MEMS-Based IMU Drift Minimization: Sage Husa Adaptive Robust Kalman Filtering. *IEEE Sens. J.* **2020**, *20*, 250–260. [CrossRef]
9. Or, B.; Bobrovsky, B.Z.; Klein, I. Kalman Filtering with Adaptive Step Size Using a Covariance-Based Criterion. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–10. [CrossRef]
10. Sarkka, S.; Nummenmaa, A. Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Trans. Autom. Control* **2009**, *54*, 596–600. [CrossRef]
11. Wu, F.; Luo, H.Y.; Jia, H.W.; Zhao, F.; Xiao, Y.M.; Gao, X.L. Predicting the Noise Covariance with a Multitask Learning Model for Kalman Filter-Based GNSS/INS Integrated Navigation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
12. Xu, H.; Yuan, H.D.; Duan, K.Q.; Xie, W.C.; Wang, Y.L. An Adaptive Gaussian Sum Kalman Filter Based on a Partial Variational Bayesian Method. *IEEE Trans. Autom. Control* **2020**, *65*, 4793–4799. [CrossRef]

13. Yasini, S.; Pelckmans, K. Worst-Case Prediction Performance Analysis of the Kalman Filter. *IEEE Trans. Autom. Control* **2018**, *63*, 1768–1775. [CrossRef]
14. Zorzi, M. Robust Kalman Filtering Under Model Perturbations. *IEEE Trans. Autom. Control* **2017**, *62*, 2902–2907. [CrossRef]
15. Morris, J.M. The Kalman filter: A robust estimator for some classes of linear quadratic problems. *IEEE Trans. Inf. Theory* **1976**, *22*, 526–534. [CrossRef]
16. Shmaliy, Y.S. An iterative Kalman-like algorithm ignoring noise and initial conditions. *IEEE Trans. Signal Process.* **2011**, *59*, 2465–2473. [CrossRef]
17. Shmaliy, Y.S.; Lehmann, F.; Zhao, S.; Ahn, C.K. Comparing Robustness of the Kalman, H_∞, and UFIR Filters. *IEEE Trans. Signal Process.* **2018**, *66*, 3447–3458. [CrossRef]
18. Shmaliy, Y.S. Linear optimal FIR estimation of discrete time-invariant state-space models. *IEEE Trans. Signal Process.* **2010**, *58*, 3086–3096. [CrossRef]
19. Zhao, S.Y.; Shmaliy, Y.S.; Liu, F. Fast Kalman-Like Optimal Unbiased FIR Filtering with Applications. *IEEE Trans. Signal Process.* **2016**, *64*, 2284–2297. [CrossRef]
20. Dehghannasiri, R.; Esfahani, M.S.; Dougherty, E.R. Intrinsically Bayesian robust Kalman filter: An innovation process approach. *IEEE Trans. Signal Process.* **2017**, *65*, 2531–2546. [CrossRef]
21. Dehghannasiri, R.; Esfahani, M.S.; Qian, X.N.; Dougherty, E.R. Optimal Bayesian Kalman Filtering with Prior Update. *IEEE Trans. Signal Process.* **2018**, *66*, 1982–1996. [CrossRef]
22. Lei, X.; Li, J. An Adaptive Altitude Information Fusion Method for Autonomous Landing Processes of Small Unmanned Aerial Rotorcraft. *Sensors* **2012**, *12*, 13212–13224. [CrossRef]
23. Xu, G.; Huang, Y.; Gao, Z.; Zhang, Y. A Computationally Efficient Variational Adaptive Kalman Filter for Transfer Alignment. *IEEE Sens. J.* **2020**, *20*, 13682–13693. [CrossRef]
24. Shan, C.; Zhou, W.; Yang, Y.; Jiang, Z. Multi-Fading Factor and Updated Monitoring Strategy Adaptive Kalman Filter-Based Variational Bayesian. *Sensors* **2021**, *21*, 198. [CrossRef] [PubMed]
25. Huang, Y.; Zhang, Y.; Wu, Z.; Li, N.; Chambers, J. A Novel Adaptive Kalman Filter with Inaccurate Process and Measurement Noise Covariance Matrices. *IEEE Trans. Autom. Control* **2018**, *63*, 594–601. [CrossRef]
26. Zhang, Y.; Jia, G.; Li, N.; Bai, M. A Novel Adaptive Kalman Filter with Colored Measurement Noise. *IEEE Access* **2018**, *6*, 74569–74578. [CrossRef]
27. Kailath, T. An innovations approach to least-squares estimation—Part I: Linear filtering in additive white noise. *IEEE Trans. Autom. Control* **1968**, *AC-13*, 646–655. [CrossRef]
28. Zadeh, L.A.; Ragazzini, J.R. An extension of Wiener's theory of prediction. *J. Appl. Phys.* **1950**, *21*, 645–655. [CrossRef]
29. Bode, H.W.; Shannon, C.E. A simplified derivation of linear least square smoothing and prediction theory. *Proc. IRE* **1950**, *38*, 417–425. [CrossRef]
30. Poor, H.V. On robust Wiener filtering. *IEEE Trans. Autom. Control* **1980**, *25*, 531–536. [CrossRef]
31. Vastola, K.S.; Poor, H.V. Robust Wiener-Kolmogorov theory. *IEEE Trans. Inf. Theory* **1984**, *30*, 316–327. [CrossRef]
32. Poor, H.V. Robust matched filters. *IEEE Trans. Inf. Theory* **1983**, *29*, 677–687. [CrossRef]
33. Ruiz, F.J.; Titsias, M.; Blei, D.M. Overdispersed Black-Box Variational Inference. In Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, Jersey City, NJ, USA, 25–29 June 2016; pp. 647–656.
34. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.