

Article

# Self-Tuning Process Noise in Variational Bayesian Adaptive Kalman Filter for Target Tracking

Yan Cheng, Shengkang Zhang \*, Xueyun Wang and Haifeng Wang

Science and Technology on Metrology and Calibration Laboratory, Beijing Institute of Radio Metrology and Measurement, Beijing 100854, China; chengyan178@163.com (Y.C.); xywang0130@163.com (X.W.); haifengwang203@163.com (H.W.)

\* Correspondence: zhangsk@126.com

**Abstract:** Many practical systems, such as target tracking, navigation systems, autonomous vehicles, and other applications, are usually applied in dynamic conditions. Thus, the actual noise statistics characteristics of these systems are generally time varying and unknown, which will deteriorate the state estimation accuracy of the Kalman filter (KF) and even cause filter diverging. To address this issue, this paper proposes an adaptive process noise covariance ( $Q_k$ )-based variational Bayesian adaptive Kalman filter (AQ-VBAKF) algorithm. Firstly, the adaptive factor is introduced to self-tune the process noise covariance; the adaptive factor is obtained based on the innovation sequences, which can adapt to the input measurement values. Then, the VB solution is applied to approximate the time variant and unknown measurement noise covariance. Therefore, this proposed algorithm can adjust the process noise covariance and the measurement noise covariance simultaneously based on the variable input signals, which can improve the self-adaptive ability of the state estimation filter in dynamic conditions. According to the dynamic target tracking test results, the proposed AQ-VBAKF outperforms several other existing filtering methods in estimation accuracy, robustness, and computational efficiency.

**Keywords:** state estimation; variational Bayesian; noise covariance matrix; adaptive factors; adaptive Kalman filter

**Citation:** Cheng, Y.; Zhang, S.; Wang, X.; Wang, H. Self-Tuning Process Noise in Variational Bayesian Adaptive Kalman Filter for Target Tracking. *Electronics* **2023**, *12*, 3887. <https://doi.org/10.3390/electronics12183887>

Academic Editor: Ryszard Sroka

Received: 20 August 2023

Revised: 11 September 2023

Accepted: 12 September 2023

Published: 14 September 2023



**Copyright:** © 2023 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

In many practical applications, such as target tracking, positioning, autonomous vehicles, and so on, state estimation has been widely deployed [1–4]. The KF (Kalman filter) can obtain the optimal estimation for the minimum mean square error (MMSE) if the noise covariances are accurate under the linear Gaussian condition [5]. In the KF iteration, noise covariances are preselected and kept constant throughout the whole filtering process. However, when the practical systems work in dynamic environments, the constant noise covariances cannot reflect the real dynamic circumstances. For example, autonomous vehicle systems [6] inherently work in dynamic environments, where the sensor measurement, environment conditions, and vehicle dynamics are time varying and difficult to obtain. In this condition, the accurate noise statistics of practical systems are usually unknown. The actual noise covariances vary with the changes in environments. Thus, constant noise covariances for KF may degrade the filtering performance in dynamic conditions. Therefore, advanced estimation methods are necessary for autonomous vehicle systems, target tracking systems, and other applications.

Accordingly, to address this issue, several adaptive KF (AKF) methods have been developed. The AKF algorithms include the multiple mode AKF [7], the Sage-Husa AKF [8], the fading AKF [9], and the variational Bayesian-based AKF (VBAKF) [10]. The multiple mode AKF requires a bank of KFs operating simultaneously, resulting in a large computational burden. The Sage-Husa AKF uses the covariance matching method to estimate

the maximum posterior noise, which cannot ensure that it will converge to the accurate noise covariances. The fading AKF introduces the fading factor to decrease the weight of the current measurement value. However, the fading AKF usually needs multiple fading factors, which are not easy to calculate [11].

The variational Bayesian-based AKF (VBAKF) has been widely utilized, as it can use a factorized free-form distribution to approximate the unknown measurement noise covariance matrix (MNCM) and the state vector's joint posterior distribution [12,13]. However, in the VBAKF algorithm, the process noise covariance matrix (PNCM) is predefined and kept constant, which cannot reflect the actual noise environment and, thus, will deteriorate the filtering performance in time-varying situations. Huang et al. [14] proposed the VBAKF-PR algorithm to estimate the online MNCM and the predicted state error covariance matrix together, and more accurate results were obtained. However, the VBAKF-PR approach needs a predefined PNCM and has a large computational burden. In reference [15], the predicted state error covariance matrix was estimated, but the PNCM stayed constant. In reference [16], to overcome the issue that the PNCM's distribution and the system state vector's distribution are non-conjugate, black-box variational inference was used to solve this problem. In [17], a sliding window variational AKF was proposed, which relies on approximating the smoothing posterior distribution. However, these algorithms either have a high computational burden or the PNCM stays constant, which will decrease the filtering performance in dynamic conditions. Since the inaccurate predefined PNCM is used for each iteration, this may cause a decline in estimation accuracy.

To enhance the VBAKF's performance in dynamic systems, an adaptive PNCM ( $\mathbf{Q}_k$ )-based variational Bayesian adaptive KF (AQ-VBAKF) method is proposed in this paper. The main contributions of this paper are as follows:

1. To further improve the VBAKF algorithm in dynamic systems and to improve the adaptability of PNCM, the proposed AQ-VBAKF algorithm can not only estimate the MNCM online, but can also self-tune the PNCM in real time. Therefore, the AQ-VBAKF can adapt to the input variable signals in the linear dynamic system.
2. The VBAKF algorithm can only estimate the MNCM, but the PNCM stays constant during the whole filtering process, which will decrease the filtering performance. In the proposed algorithm, the adaptive factor is introduced to tune the PNCM in real time, meaning the fixed PNCM does not cause a decline in the VBAKF estimation accuracy at each iteration.
3. The proposed algorithm is evaluated using the target tracking simulation. Based on a series of experiments, the proposed AQ-VBAKF can improve the filtering accuracy and robustness in comparison with other filtering methods under dynamic conditions. Moreover, the computational burden is not heavy, thus it can be applied in practical systems.

The structure of the paper is summarized as follows. The state-space model and problem statement are described in Section 2. The proposed AQ-VBAKF approach is illustrated in Section 3. The AQ-VBAKF method is verified by target tracking tests in Section 4. The conclusions are drawn in Section 5.

## 2. State-Space Model and Problem Statement

For the linear Gaussian discrete-time system, the state-space model is demonstrated as follows:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \boldsymbol{\omega}_{k-1} \quad (1)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (2)$$

where Equation (1) denotes the system model, and Equation (2) denotes the measurement model.  $k \in \mathbb{N}$  represents the sampling time point.  $\mathbf{x}_k \in \mathbb{R}^{n_x}$  and  $\mathbf{z}_k \in \mathbb{R}^{n_z}$  denote the state vector with  $n_x$  dimensions and the measurement vector with  $n_z$  dimensions at the

$k$ -th epoch, respectively.  $\mathbf{F}_k$  and  $\mathbf{H}_k$  are the system transition matrix and the observation matrix at the  $k$ -th epoch. The dimensions of  $\mathbf{F}_k$  and  $\mathbf{H}_k$  are  $n_x \times n_x$  and  $n_z \times n_x$ , respectively.  $\boldsymbol{\omega}_{k-1}$  is the process noise that obeys  $N(0, \mathbf{Q}_{k-1})$ .  $\mathbf{v}_k$  represents the measurement noise that obeys  $N(0, \mathbf{R}_k)$ . The dimensions of  $\mathbf{Q}_{k-1}$  and  $\mathbf{R}_k$  are  $n_x \times n_x$  and  $n_z \times n_z$ , respectively.

The KF algorithm includes two procedures, which are the prediction process and the correction process. The prediction procedure is expressed as [18]:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \tag{3}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1} \tag{4}$$

Then, the correction process is defined as [18]:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_{k-1})^{-1} \tag{5}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \tag{6}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{7}$$

where  $\mathbf{K}_k$  is the KF gain. The dimension of  $\mathbf{K}_k$  is  $n_x \times n_z$ .  $\hat{\mathbf{x}}_{k|k-1}$  and  $\hat{\mathbf{x}}_{k-1|k-1}$  represent the predicted and estimated state values.  $\mathbf{P}_{k|k-1}$  and  $\mathbf{P}_{k-1|k-1}$  represent the predicted and estimated state error covariance matrices.

In dynamic conditions, such as integration navigation systems, target tracking systems, and autonomous vehicles, the actual noise covariances vary with the changes in environments and the accurate noise statistics of practical systems are usually unknown. The inaccurate and constant noise covariances  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  in the KF may cause suboptimal estimations and even divergence. Therefore, an adaptive KF algorithm for the inaccurate noise covariance matrices is necessary under practical dynamic conditions. Accordingly, a novel AQ-VBAKF approach is proposed to self-tune the variable and unknown MNMCM and PNCM under dynamic conditions.

### 3. Proposed AQ-VBAKF Algorithm

#### 3.1. Variational Bayesian Approximation for MNMCM ( $\mathbf{R}_k$ )

According to Equations (1) and (2), the probability form for the state-space model is [19]:

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{R}_k) = N(\mathbf{z}_k | \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k) \tag{8}$$

where  $N(\cdot | \boldsymbol{\mu}, \boldsymbol{\Sigma})$  represents the multivariate Gaussian distribution with a mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ . Because  $\mathbf{R}_k$  is unknown, the VB algorithm is applied to acquire the posterior distribution  $p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k})$ . However, it is usually not analytical to handle [12]. The approximation scheme VB is used to approximate the intractable posterior inference in the Bayesian model [12].

In linear and Gaussian systems,  $\mathbf{x}_k$  is considered to be independent of the NCM. The joint distribution at the  $(k-1)$ -th epoch  $p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k-1})$  is decomposed as:

$$p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) p(\mathbf{R}_k | \mathbf{z}_{1:k-1}) \tag{9}$$

$p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$  is assumed as the Gaussian distribution, and  $p(\mathbf{R}_k | \mathbf{z}_{1:k-1})$  can be considered as an inverse Wishart (IW) distribution, which is expressed as [16]:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = N(\mathbf{x}_k | \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{10}$$

$$p(\mathbf{R}_k | \mathbf{z}_{1:k-1}) = IW(\mathbf{R}_k | \hat{\mathbf{v}}_{k|k-1}, \hat{\mathbf{V}}_{k|k-1}) \tag{11}$$

where  $IW(\cdot | w, \mathbf{W})$  represents the IW distribution.  $w$  denotes the degrees of freedom's number.  $\mathbf{W}$  represents the inverse scale matrix.

Employing the VB inference, the joint posterior distribution can be approximated as:

$$p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k}) \approx q(\mathbf{x}_k)q(\mathbf{R}_k) \tag{12}$$

where  $q(\cdot)$  is the approximation for  $p(\cdot)$ .  $q(\mathbf{x}_k)$  and  $q(\mathbf{R}_k)$  are obtained based on the minimization of the Kullback-Leibler (KL) divergence between the real posterior distribution  $p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k})$  and the approximation  $q(\mathbf{x}_k)q(\mathbf{R}_k)$ , which is:

$$\begin{aligned} q(\mathbf{x}_k)q(\mathbf{R}_k) &= \arg \min \{ \text{KL}(q(\mathbf{x}_k)q(\mathbf{R}_k) \| p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k})) \} \\ &= \arg \min \left\{ \int q(\mathbf{x}_k)q(\mathbf{R}_k) \log \left( \frac{q(\mathbf{x}_k)q(\mathbf{R}_k)}{p(\mathbf{x}_k, \mathbf{R}_k | \mathbf{z}_{1:k})} \right) d\mathbf{x}_k d\mathbf{R}_k \right\} \end{aligned} \tag{13}$$

where  $\text{KL}(m(x)||n(x))$  represents the KL divergence between  $m(x)$  and  $n(x)$ .

The optimal solution of Equation (13) can be obtained as follows [20]:

$$\log q(\mathbf{R}_k) = E_{\mathbf{x}_k}[\log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_{1:k})] + C_{\mathbf{R}_k} \tag{14}$$

$$\log q(\mathbf{x}_k) = E_{\mathbf{R}_k}[\log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_{1:k})] + C_{\mathbf{x}_k} \tag{15}$$

where  $E[\cdot]$  represents the expectation operator.  $C_{\mathbf{R}_k}$  and  $C_{\mathbf{x}_k}$  denote the constants independent of  $\mathbf{R}_k$  and  $\mathbf{x}_k$ , respectively.

Because of the Bayesian rule, it can be given that:

$$\log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_{1:k}) = \log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1}) + \log p(\mathbf{z}_{1:k-1}) \tag{16}$$

Due to  $p(\mathbf{z}_{1:k-1})$  being independent of  $\mathbf{R}_k$  and  $\mathbf{x}_k$ , then:

$$\log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_{1:k}) \propto \log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1}) \tag{17}$$

Based on Equations (8), (10), and (11),  $p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$  is decomposed as [21]:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1}) &= p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{R}_k) p(\mathbf{x}_k | \mathbf{R}_k, \mathbf{z}_{1:k-1}) p(\mathbf{R}_k | \mathbf{z}_{1:k-1}) \\ &= \mathbf{N}(\mathbf{z}_k | \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k) \mathbf{N}(\mathbf{x}_k | \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) IW(\mathbf{R}_k | \hat{\mathbf{V}}_{k|k-1}, \hat{\mathbf{V}}_{k|k-1}) \end{aligned} \tag{18}$$

Accordingly, we can have:

$$\begin{aligned} \log p(\mathbf{x}_k, \mathbf{R}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1}) &= \\ &= -0.5(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k)^T \mathbf{R}_k^{-1} (\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k) \\ &= -0.5(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T \mathbf{P}_{k|k-1}^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \\ &= -0.5(\hat{\mathbf{V}}_{k|k-1} + n_z + 2) \log |\mathbf{R}_k| - 0.5 \text{tr}(\hat{\mathbf{V}}_{k|k-1} \mathbf{R}_k^{-1}) + C \end{aligned} \tag{19}$$

where  $\text{tr}[\cdot]$  denotes the matrix's trace.  $n_x$  and  $n_z$  denote the dimensions of  $\mathbf{x}_k$  and  $\mathbf{z}_k$ , respectively.  $C$  represents a constant independent of  $\mathbf{x}_k$  and  $\mathbf{R}_k$ .

Substituting Equations (17) and (19) into Equation (14), we can obtain the updated  $q(\mathbf{R}_k)$ .  $q(\mathbf{R}_k)$  is approximated as the IW distribution:

$$q^{(s)}(\mathbf{R}_k) = IW(\mathbf{R}_k | \hat{\mathbf{V}}_{k|k}^{(s)}, \hat{\mathbf{V}}_{k|k}^{(s)}) \tag{20}$$

where

$$\hat{\mathbf{V}}_{k|k}^{(s)} = \hat{\mathbf{V}}_{k|k-1} + 1 \tag{21}$$

$$\hat{\mathbf{V}}_{k|k}^{(s)} = \mathbf{N}_k^{(s-1)} + \hat{\mathbf{V}}_{k|k-1} \tag{22}$$

$$\mathbf{N}_k^{(s-1)} = (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}^{(s-1)})(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}^{(s-1)})^T + \mathbf{H}_k \mathbf{P}_{k|k}^{(s-1)} \mathbf{H}_k^T \tag{23}$$

The superscript  $s$  denotes the  $s$ -th VB iteration. The dimensions of  $\hat{\mathbf{V}}_{k|k-1}$  is  $n_z \times n_z$  and  $\hat{v}_{k|k-1}$  is scalar.

Substituting Equations (17) and (19) into Equation (15),  $q(\mathbf{x}_k)$  is updated as:

$$q^{(s)}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k | \hat{\mathbf{x}}_{k|k}^{(s)}, \mathbf{P}_{k|k}^{(s)}) \tag{24}$$

where

$$\mathbf{K}_k^{(s)} = \hat{\mathbf{P}}_{k|k-1}^{(s)} \mathbf{H}_k^T \left[ \mathbf{H}_k \hat{\mathbf{P}}_{k|k-1}^{(s)} \mathbf{H}_k^T + \left( \mathbf{E}^{(s)} [\mathbf{R}_k^{-1}] \right)^{-1} \right]^{-1} \tag{25}$$

$$\hat{\mathbf{x}}_{k|k}^{(s)} = \hat{\mathbf{x}}_{k|k-1}^{(s)} + \mathbf{K}_k^{(s)} (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{(s)}) \tag{26}$$

$$\mathbf{P}_{k|k}^{(s)} = \hat{\mathbf{P}}_{k|k-1}^{(s)} - \mathbf{K}_k^{(s)} \mathbf{H}_k \hat{\mathbf{P}}_{k|k-1}^{(s)} \tag{27}$$

### 3.2. Adaptation of PNCM ( $\mathbf{Q}_k$ )

In the VBAKF, the MNMCM ( $\mathbf{R}_k$ ) is adjusted in real time. However, the PNCM ( $\mathbf{Q}_k$ ) stays constant in the filtering process. The constant  $\mathbf{Q}_k$  cannot match the actual noise statistic characteristic, so the adaptive method of  $\mathbf{Q}_k$  is proposed to enhance the estimation accuracy. Hence, the adaptive factor  $\lambda_k$  is introduced to adjust  $\mathbf{Q}_k$ , that is:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \lambda_k \mathbf{Q}_{k-1} \tag{28}$$

In the KF estimation, when the state estimations are accurate, the following equation is held [22]:

$$\begin{aligned} E[\mathbf{d}_{k+j} \mathbf{d}_k^T] &\approx \\ \mathbf{H}_k \mathbf{F}_k (\mathbf{I} - \mathbf{K}_{k+j-1} \mathbf{H}_k) \cdot \mathbf{F}_k (\mathbf{I} - \mathbf{K}_{k+2} \mathbf{H}_k) \cdot \mathbf{F}_k (\mathbf{P}_{k|k-1} \mathbf{H}_k^T - \mathbf{K}_k \mathbf{C}_k) \quad j=1, 2, \dots \end{aligned} \tag{29}$$

where  $\mathbf{I}$  represents the identity matrix and  $\mathbf{d}_k$  denotes innovation sequences, that is:

$$\mathbf{d}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{30}$$

where  $\mathbf{C}_k$  is the innovation covariance. The dimensions of  $\mathbf{d}_k$  and  $\mathbf{C}_k$  are  $n_z \times 1$  and  $n_z \times n_z$ , respectively. To keep the estimation accurate,  $E[\mathbf{d}_{k+j} \mathbf{d}_k^T] = 0$  should be satisfied [22]. Therefore, according to Equation (29),  $\mathbf{K}_k$  should be selected to establish the following equation:

$$\mathbf{P}_{k|k-1} \mathbf{H}_k^T - \mathbf{K}_k \mathbf{C}_k = 0 \tag{31}$$

By substituting Equation (5) into Equation (31), it can be given that:

$$\mathbf{P}_{k|k-1} \mathbf{H}_k^T \left\{ \mathbf{I} - \left( \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_{k-1} \right)^{-1} \mathbf{C}_k \right\} = 0 \tag{32}$$

To satisfy Equation (32), the following equation should be held:

$$\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_{k-1} = \mathbf{C}_k \tag{33}$$

Because of Equation (28), then it can then be determined that:

$$\mathbf{C}_k = \mathbf{H}_k \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \mathbf{H}_k^T + \mathbf{R}_{k-1} + \lambda_k \mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T \tag{34}$$

Therefore, the adaptive factor  $\lambda_k$  can be obtained as:

$$\lambda_k = \frac{\text{tr}(\mathbf{C}_k - \mathbf{H}_k \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \mathbf{H}_k^T - \mathbf{R}_{k-1})}{\text{tr}(\mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T)} \tag{35}$$

To keep the whole filtering process stable, the adaptive factor  $\lambda_k$  should be larger than 1; that is:

$$\lambda_k = \max(1, \lambda_k) \tag{36}$$

When the system experiences sudden changes, and the adaptive factor is larger than 1, then the corresponding Kalman gain will become larger based on Equation (5). Therefore, according to Equation (6), more weight will be added to the current measurement data and less weight will be given to the past state information, and then the state estimation will be more accurate. Therefore,  $\lambda_k$  is obtained with:

$$\lambda_k = \begin{cases} \frac{\text{tr}(\mathbf{C}_k - \mathbf{H}_k \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \mathbf{H}_k^T - \mathbf{R}_{k-1})}{\text{tr}(\mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T)} & \lambda_k \geq 1 \\ 1 & \lambda_k < 1 \end{cases} \quad (37)$$

The innovation covariance  $\mathbf{C}_k$  is defined as:

$$\mathbf{C}_k = \text{E}[\mathbf{d}_k \mathbf{d}_k^T] \quad (38)$$

In general, it is difficult to obtain the actual innovation covariance. Thus, we use the window function method to estimate the innovation covariance [23]:

$$\hat{\mathbf{C}}_k = \hat{\mathbf{C}}_{k-1} + \frac{1}{N} (\mathbf{d}_k \mathbf{d}_k^T - \mathbf{d}_{k-N} \mathbf{d}_{k-N}^T) \quad (39)$$

where  $\hat{\mathbf{C}}_k$  denotes the estimated innovation covariance, and  $N > 1$  represents the window size and is a scalar.

### 3.3. Proposed Algorithm

To summarize, the structure diagram of the proposed algorithm is shown in Figure 1. The detailed algorithm is demonstrated in Algorithm 1, where  $\xi \in (0, 1]$  represents the forgetting factor and  $S$  denotes the VB iteration number.

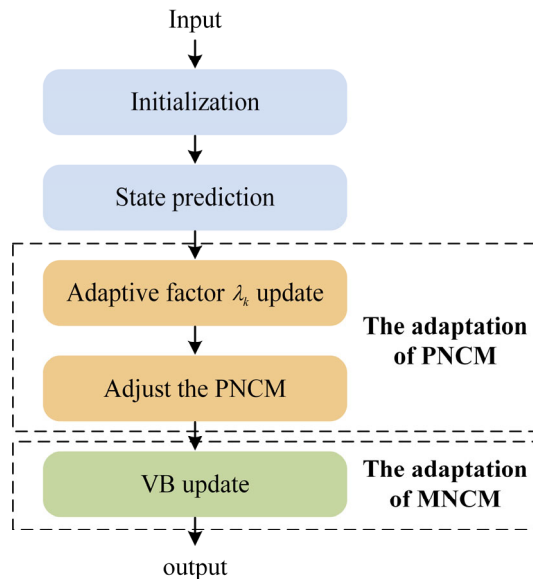


Figure 1. Structure diagram of the proposed AQ-VBAKF algorithm.

---

#### Algorithm 1: Proposed AQ-VBAKF Algorithm

---

**Input:**  $\hat{\mathbf{x}}_{k-1|k-1}$ ,  $\mathbf{z}_k$ ,  $\mathbf{P}_{k-1|k-1}$ ,  $\mathbf{R}_{k-1}$ ,  $\mathbf{Q}_{k-1}$ ,  $\hat{\mathbf{V}}_{k-1|k-1}$ ,  $\hat{\mathbf{v}}_{k-1|k-1}$ ,  $\mathbf{F}_k$ ,  $\mathbf{H}_k$ ,  $N$ ,  $n_z$ ,  $S$ ,  $\xi$

**Time update for state values:**

1:  $\hat{\mathbf{x}}_{k|k-1}$  can be obtained via Equation (3).

**The adaptive factor update and the time update for  $\mathbf{P}_{k|k-1}$ :**

---

2: The adaptive factor is computed using Equations (30), (37), and (39).

3:  $\mathbf{P}_{k|k-1}$  is obtained with Equation (28).

**VB estimation update:**

4: Initialization:  $\hat{\mathbf{P}}_{k|k-1}^{(0)} = \mathbf{P}_{k|k-1}$ ,  $\hat{\mathbf{x}}_{k|k-1}^{(0)} = \hat{\mathbf{x}}_{k|k-1}$

$$\hat{\mathbf{v}}_{k|k-1} = \xi(\hat{\mathbf{v}}_{k-1|k-1} - n_z - 1) + n_z + 1, \quad \hat{\mathbf{V}}_{k|k-1} = \xi \hat{\mathbf{V}}_{k-1|k-1}$$

For  $s = 1: S$

5: Update  $q^{(s)}(\mathbf{R}_k)$  using Equations (21)–(23)

6:  $(\mathbf{E}^{(s)}[\mathbf{R}_k^{-1}])^{-1}$  is obtained with  $(\mathbf{E}^{(s)}[\mathbf{R}_k^{-1}])^{-1} = (\hat{\mathbf{v}}_{k|k}^{(s)} - n_z - 1)^{-1}(\hat{\mathbf{V}}_k^{(s)})$

7:  $\hat{\mathbf{x}}_{k|k}^{(s)}$  and  $\hat{\mathbf{P}}_{k|k}^{(s)}$  are obtained via Equations (25)–(27)

End

**Output:**  $\mathbf{P}_{k|k}$ ,  $\hat{\mathbf{x}}_{k|k}$ ,  $\hat{\mathbf{v}}_{k|k}$ ,  $\hat{\mathbf{V}}_{k|k}$

**4. Simulation Verification and Analysis**

In this section, we adopt a target tracking test to assess the availability and superiority of the proposed AQ-VBAKF algorithm. The target tracking example’s state equation and measurement equation are:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{I}_2 & \Delta\sigma\mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{n}_{k-1} \tag{40}$$

$$\mathbf{z}_k = [\mathbf{I}_2 \quad \mathbf{0}] \mathbf{x}_{k-1} + \mathbf{v}_k \tag{41}$$

where  $\mathbf{x}_k = [x_k \quad y_k \quad \dot{x}_k \quad \dot{y}_k]^T$  is the system state vector. Then,  $(x_k, y_k)$  and  $(\dot{x}_k, \dot{y}_k)$  stand for the unknown positions and velocities.  $\Delta\sigma$  is the sampling interval and  $\Delta\sigma$  is set to 1 s. The true covariances of  $\mathbf{n}_{k-1}$  and  $\mathbf{v}_k$  are described by:

$$\mathbf{Q}_k = [0.5 \cos(\frac{\pi k}{T}) + 6.5] \begin{bmatrix} \frac{\Delta\sigma^3}{3} \mathbf{I}_2 & 0.5\Delta\sigma^2 \mathbf{I}_2 \\ 0.5\Delta\sigma^2 \mathbf{I}_2 & \Delta\sigma \mathbf{I}_2 \end{bmatrix} q \tag{42}$$

$$\mathbf{R}_k = [0.05 \cos(\frac{\pi k}{T}) + 0.1] \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} r \tag{43}$$

where  $\mathbf{I}_2$  is a two-dimensional identity matrix.  $T$  denotes the simulation time and  $T = 1000$  s. Additionally,  $q$  is set as 1 m<sup>2</sup>/s<sup>3</sup> and  $r$  is set as 100 m<sup>2</sup>. Moreover, the initial values for PNCM ( $\mathbf{Q}_0$ ) and MNCM ( $\mathbf{R}_0$ ) are set to  $\mathbf{Q}_0 = \alpha \mathbf{I}_4$  and  $\mathbf{R}_0 = \beta \mathbf{I}_2$ .

*4.1. Estimation Performance Analysis*

To verify the filtering accuracy, we compare the proposed AQ-VBAKF with five other filtering estimation methods:

1. KFTCM algorithm: the KF is iterated with true covariance matrices (KFTCM), which act as a reference.
2. KFNCM algorithm: the KF is iterated with nominal covariance matrices (KFNCM), which are  $\mathbf{Q}_0$  and  $\mathbf{R}_0$ .  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are fixed during the filtering process.
3. AQKF algorithm: the KF is iterated with the adaptive  $\mathbf{Q}_k$ ; only  $\mathbf{Q}_k$  is adjusted based on the previous Section 3.2, and  $\mathbf{R}_k$  is fixed throughout the whole process.
4. VBKF algorithm:  $\mathbf{R}_k$  is updated using the VB method and  $\mathbf{Q}_k$  is constant throughout the whole process.

5. VBAKF-PR algorithm: the VBAKF-PR algorithm was demonstrated in [14].

In the simulation, the initial values are set to  $\mathbf{P}_{00} = \text{diag}([100, 100, 100, 100])$  and  $\mathbf{x}_0 = [100, 100, 10, 10]^T$ . Table 1 shows the different algorithms' parameter settings.  $S$  is the VB iteration times. The iteration times should be proper to ensure the VB approximate converges to the optimal results. Generally, VB iteration times should be more than 6 to make the filter converge, thus  $S$  is chosen as 15 [24].  $N$  represents the window size for the AQKF and AQ-VBAKF algorithms, which is analyzed in the simulation later.  $\alpha$  and  $\beta$  are used for setting  $\mathbf{Q}_0 = \alpha \mathbf{I}_4$  and  $\mathbf{R}_0 = \beta \mathbf{I}_2$ . In fact, the prior noise information of the system is usually unknown; thus, the setting of  $\mathbf{Q}_0$  and  $\mathbf{R}_0$  are usually not accurate. Moreover,  $\alpha$  and  $\beta$  are both variables to test the robustness of the algorithms with different  $\mathbf{Q}_0$  and  $\mathbf{R}_0$  values.  $\xi$  is used for VB inference initialization, as shown in Algorithm 1, which should be  $\xi \in (0, 1]$  to ensure that the posterior and prior PDFs have the same forms.  $\tau$  is used for the VBAKF-PR algorithm.

**Table 1.** Parameters settings for different algorithms.

Algorithms	$\alpha$	$\beta$	$\xi$	$S$	$N$	$\tau$
KFNCM	1	10	-	-	-	-
AQKF	1	10	-	-	5	-
VBAKF	1	10	0.99	15	-	-
VBAKF-PR	1	10	0.99	15	-	10
Proposed AQ-VBAKF	1	10	0.99	15	5	-

The RMSE (root mean square error) and ARMSE (average RMSE) are used to validate the state estimation accuracy based on the following formulas:

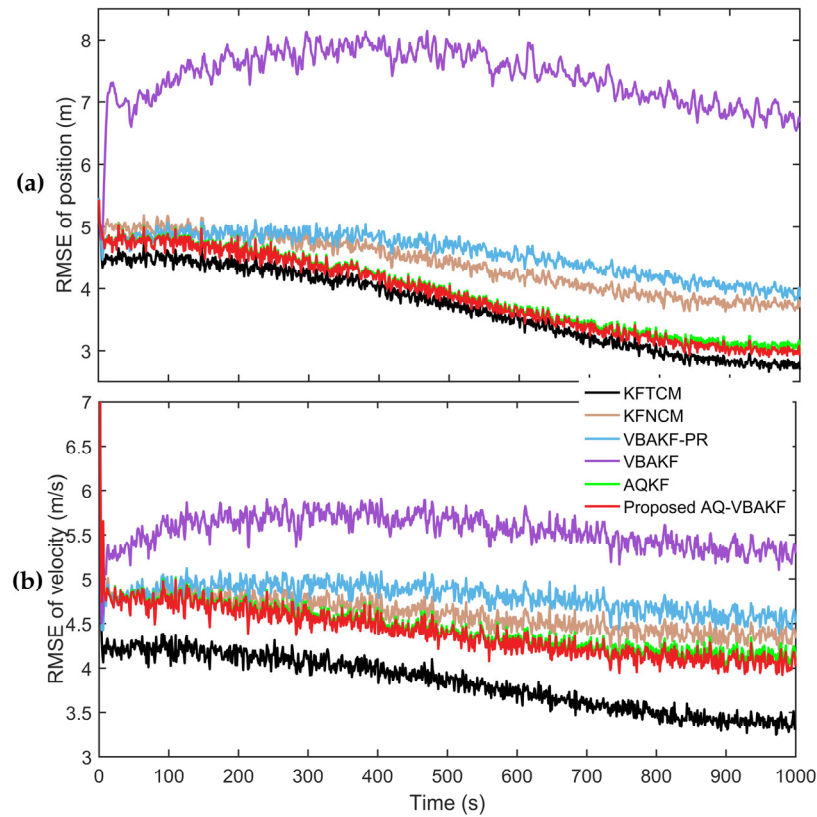
$$\text{RMSE}_{\text{pos}}(j) \triangleq \sqrt{\frac{1}{M_c} \sum_{j=1}^{M_c} [(x_j^l - \hat{x}_j^l)^2 - (y_j^l - \hat{y}_j^l)^2]} \quad (44)$$

$$\text{ARMSE}_{\text{pos}} \triangleq \frac{1}{T} \sum_{j=1}^T \text{RMSE}_{\text{pos}}(j) \quad (45)$$

where  $M_c$  represents the Monte Carlo (MC) test number, and  $M_c = 1000$ .  $(x_i^l, y_i^l)$  and  $(\hat{x}_i^l, \hat{y}_i^l)$  represent the true and the estimated positions at the  $l$ -th MC test, respectively.  $T$  is set to 1000 s. For velocity, the definitions of RMSE and ARMSE are similar.

Figure 2 demonstrates the RMSEs for different algorithms, where the KFNCM has the smallest RMSEs, and acts as a reference. In comparison with the KFNCM, AQKF, VBAKF, and VBAKF-PR algorithms, the proposed AQ-VBAKF has the smallest RMSEs for position and velocity. The VBAKF has the worst performance, which indicates that the VBAKF is even worse than the KFNCM when the initial values are far away from the true values. The VBAKF cannot guarantee an optimal outcome in any case. Due to the adaptive predicted state error covariance matrix, the VBAKF-PR shows higher accuracy than the VBAKF, but is also worse than the KFNCM due to the initial value deviating from the true values. The AQKF is slighter worse than the proposed AQ-VBAKF since the AQKF lacks the ability to update the MNCM ( $\mathbf{R}_k$ ).



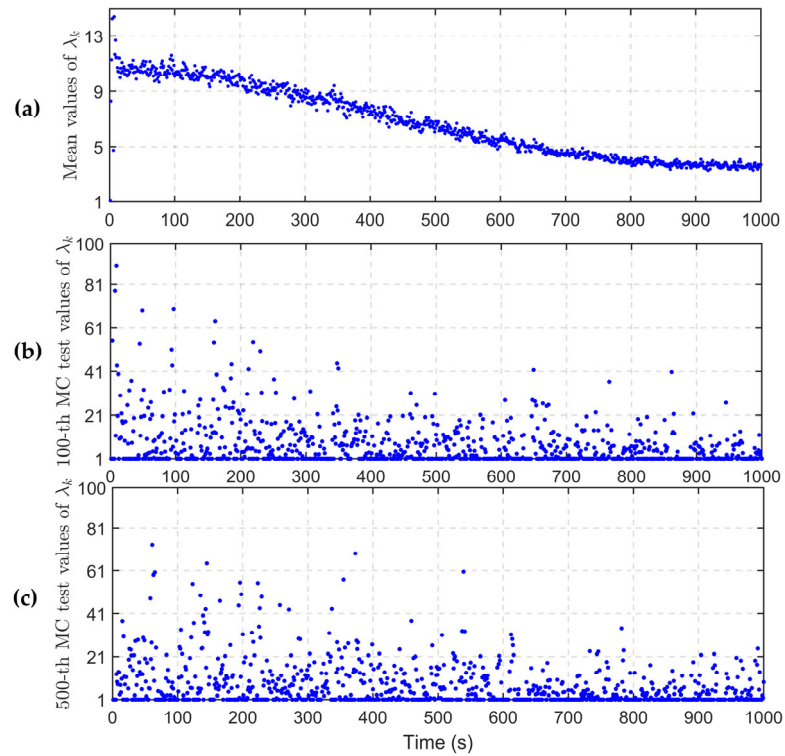


**Figure 2.** RMSEs for different filtering algorithms: (a) the RMSEs of position for different filtering algorithms; (b) the RMSEs of velocity for different filtering algorithms.

Figure 3a shows the mean values of  $\lambda_k$  in the 1000 Monte Carlo (MC) tests based on the following formulas:

$$\lambda_{\text{mean}}(k) \triangleq \frac{1}{M_c} \sum_{j=1}^{M_c} \lambda_j(k) \tag{46}$$

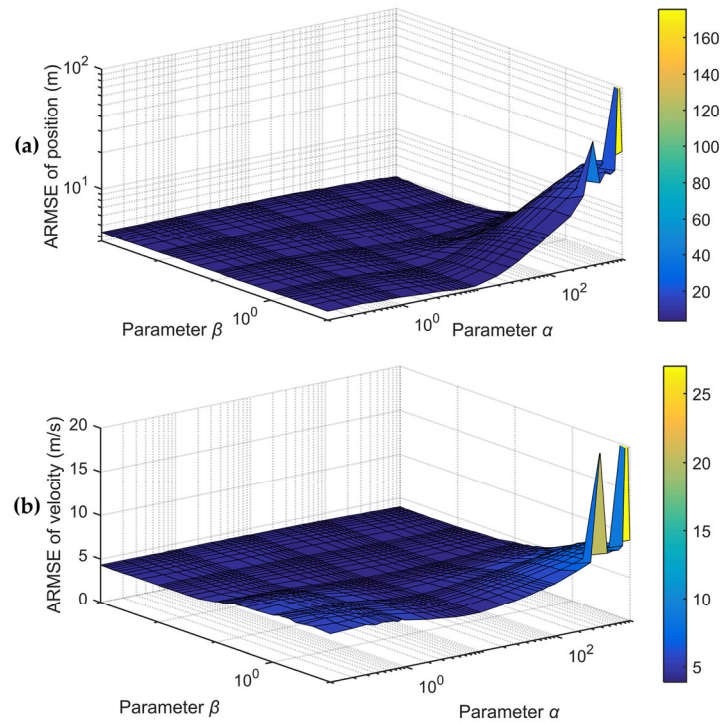
where  $M_c = 1000$ , and  $k$  denotes the  $k$ -th epoch. Figure 3b,c show the 100th and 500th MC test values of  $\lambda_k$ , respectively. It can be shown that when the RMSEs of position and velocity are higher in Figure 2 for  $k \in [0, 500]$ , the values of  $\lambda_k$  are higher in Figure 3. This indicates that when the estimations of position and velocity deviate from the true values, the values of  $\lambda_k$  will be larger to adjust the PNCM. Instead, when the estimations of position and velocity are close to the true values, the values of adaptive factor  $\lambda_k$  will become smaller. This indicates that the adaptive factor  $\lambda_k$  can adjust the PNCM according to the input signals.



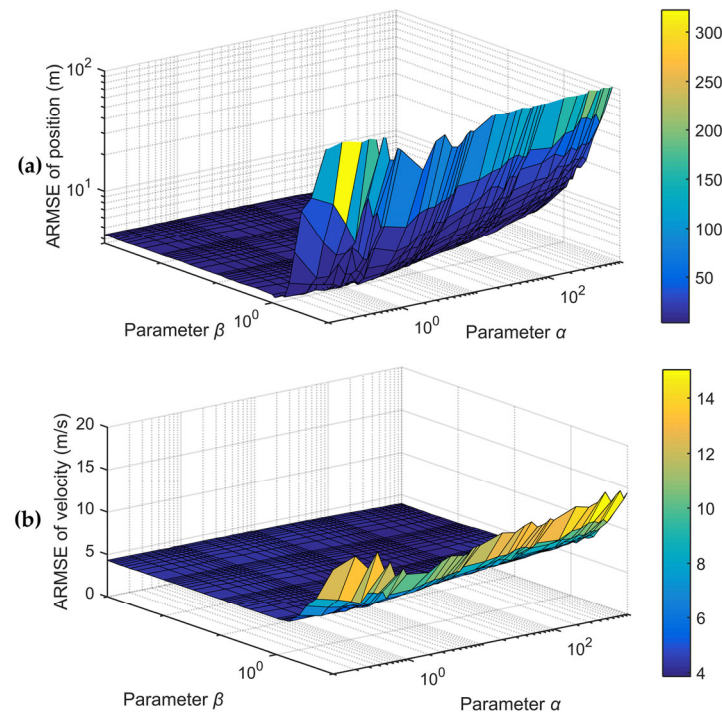
**Figure 3.** Values of  $\lambda_k$  in the proposed AQ-VBAKF algorithm: (a) the mean values of  $\lambda_k$ ; (b) the 100th MC test values of  $\lambda_k$ ; (c) the 500th MC test values of  $\lambda_k$ .

*4.2. Robustness Analysis for PNCM and MNCM Initial Value Settings*

To assess the proposed AQ-VBAKF’s robustness, the ARMSEs of the AQ-VBAKF algorithm for different initial PNCM ( $\mathbf{Q}_0$ ) and MNCM ( $\mathbf{R}_0$ ) values were calculated and are presented in Figure 4. For comparison, Figure 5 demonstrates the ARMSEs for the VBAKF-PR.  $\alpha$  and  $\beta$  are selected in the range of  $(\alpha, \beta) \in [0.1, 1000] \times [0.1, 1000]$ . To reduce the time consumption and keep the results reliable,  $M_c$  is set to 20. Moreover, Table 1 demonstrates other parameter settings, including  $\xi$ ,  $S$ ,  $N$ , and  $\tau$ .



**Figure 4.** ARMSEs of position and velocity for the AQ-VBAKF under  $(\alpha, \beta) \in [0.1, 1000] \times [0.1, 1000]$ : (a) ARMSEs of position for the proposed AQ-VBAKF; (b) ARMSEs of velocity for the proposed AQ-VBAKF.



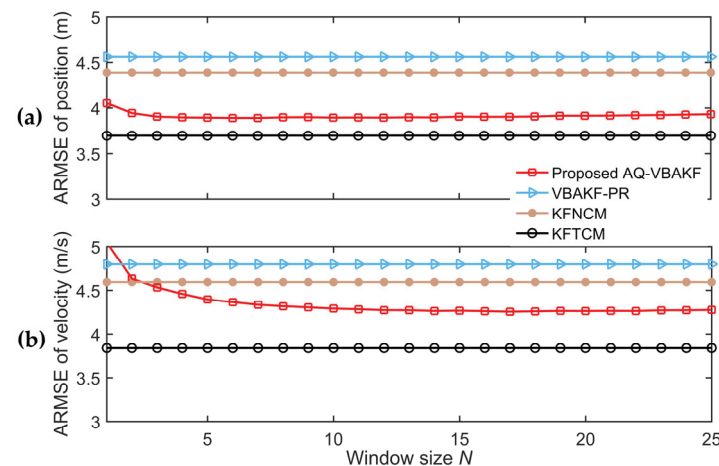
**Figure 5.** ARMSEs of position and velocity for the VBAKF-PR algorithm under  $(\alpha, \beta) \in [0.1, 1000] \times [0.1, 1000]$ : (a) ARMSEs of position for the VBAKF-PR; (b) ARMSEs of velocity for the VBAKF-PR.

In Figure 4, as we can see, the ARMSEs for position and velocity stay small and stable in most cases. However, when the parameters are in the range of  $(\alpha, \beta) \in [100, 1000] \times [0.1, 1]$ , the ARMSEs for position and velocity increase, which indicates that the filtering estimations are not so accurate. Figure 5 shows the VBAKF-PR's results. In the range of  $(\alpha, \beta) \in [0.1, 1000] \times [0.1, 1]$ , as the parameter  $\beta$  decreases, the ARMSEs increase. Therefore, the AQ-VBAKF is more robust than the VBAKF-PR in the range of  $(\alpha, \beta) \in [0.1, 100] \times [0.1, 1]$ , in which the initial values of the PNCM ( $\mathbf{Q}_0$ ) and MNCM ( $\mathbf{R}_0$ ) deviate from the true value. Therefore, when the initial values of  $\beta$  are smaller than 1, which represents that the initial MNCM ( $\mathbf{R}_0$ ) deviates from the true value, the proposed AQ-VBAKF algorithm is more robust than the VBAKF-PR.

In many practical systems, the initial state values are usually unknown. In this condition, when the initial settings of the MNCM and PNCM values differ from the true values, the proposed AQ-VBAKF performs better in self-adaptability and robustness than the VBAKF-PR algorithm.

#### 4.3. Analysis of Window Size Setting

To analyze the effect of the window size on the proposed AQ-VBAKF algorithm, some further simulations are performed. The window size is set as  $N = 1:1:25$ , and other parameters are the same as those in Table 1. The ARMSEs of different window sizes are shown in Figure 6. The KFTCM has the smallest ARMSEs of position and velocity, which acts as a reference.



**Figure 6.** ARMSEs of position and velocity under different window sizes  $N$ : (a) ARMSEs of position under different window sizes; (b) ARMSEs of velocity under different window sizes.

In Figure 6a, the ARMSEs of the position for the proposed AQ-VBAKF are smaller than that for the VBAKF-PR and KFNCM when  $N = 1:1:25$ . In Figure 6b, the ARMSEs of the position for the proposed AQ-VBAKF are smaller than that for the VBAKF-PR and KFNCM when  $N \geq 4$ . As we can see, the ARMSEs for both position and velocity cannot decrease as the window size  $N$  increases when  $N = 1:1:25$ . When the window size  $N$  is larger than 10, the ARMSEs for both the position and velocity estimations tend to be stable.

In dynamic conditions, the window size cannot be too large. A small window size can help the innovation covariance adapt to the time-varying input signals [25]. On the other hand, the window size should be large enough to enhance the innovation covariance estimation stability. Moreover, if the window size is too small, the computational burden will be heavier. Accordingly, taking into account the innovation covariance estimation accuracy and computational efficiency simultaneously in the dynamic environment, the window size  $N$  should be set as  $4 \leq N \leq 10$ .

#### 4.4. Computational Efficiency Comparison

To verify the computational efficiency, Table 2 shows the total simulation time for the different algorithms. The parameter settings are shown in Table 1, and  $M_c = 1000$ . The proposed method adds only 9% computational time compared with the VBAKF method and is only about half of the VBAKF-PR simulation time. Therefore, the proposed method has almost the same computational burden compared with the VBAKF algorithm and the computational complexity is not high.

**Table 2.** Total computational time for different algorithms.

Algorithms	Computational Time (s)
KF	37.7627
AQKF	45.0514
VBAKF	172.4772
VBAKF-PR	356.0744
Proposed AQ-VBAKF	188.0822

To further evaluate the computational complexity, the total floating-point operations of each step for the proposed AQ-VBAKF were analyzed and are shown in Table 3, where  $n_x$  and  $n_z$  are the dimensions of the state vector and measurement vector, respectively.  $S$  is the VB iteration time.

**Table 3.** Floating-point operations of proposed AQ-VBAKF algorithms.

Steps	Floating-Point Operations
1	$2n_x^2 - n_x$
2	$6n_x^2n_z + 4n_xn_z^2 + 2n_z + 5n_z^2 - 1$
3	$2n_x^3 + 2n_x^2$
4, 5, 6	$S(2n_x^2n_z + 2n_xn_z^2 + n_xn_z + n_z + 3n_z^2 + 3)$
7	$S(2n_x^3 + 6n_x^2n_z - n_x^2 + n_xn_z + 4n_xn_z^2 + n_z^2)$

According to Table 3, the total floating-point operations of the proposed AQ-VBAKF are:

$$f_{AQ-VBAKF} = 2n_x^3 + 4n_x^2 + 6n_x^2n_z - n_x + 4n_xn_z^2 + 2n_z + 5n_z^2 - 1 + S(2n_x^3 + 8n_x^2n_z - n_x^2 + 6n_xn_z^2 + 2n_xn_z + 4n_z^2 + n_z + 3) \tag{47}$$

Based on [14,17], the total floating-point operations of VBAKF-PR are:

$$f_{VBAKF-PR} = 2n_x^3 + 4n_x^2 + n_z^2 + 1 + S[n_x^3 + O(2n_x^3) + 4n_x^2n_z + 3n_xn_z^2 + O(3n_z^3) + 2n_x^2 + 4n_xn_z + 2n_z^2] \tag{48}$$

Based on Equations (47) and (48),  $f_{AQ-VBAKF}$  and  $f_{VBAKF-PR}$  are almost proportional to the VB iteration times  $S$ . Substrating Equation (48) to (47), then:

$$f_{VBAKF-PR} - f_{AQ-VBAKF} = -6n_x^2n_z + n_x - 4n_xn_z^2 - 2n_z - 4n_z^2 + 1 + S[-n_x^3 + O(2n_x^3) - 4n_x^2n_z + 3n_x^2 - 3n_xn_z^2 + 2n_xn_z + O(3n_z^3) - 2n_z^2 - n_z - 3] \tag{49}$$

According to Equation (49), with the increase in the VB iteration time  $S$ , the computational burden of VBAKF-PR is larger than the proposed AQ-VBAKF, which is consistent with the computational time results.

## 5. Conclusions

This paper proposes a novel AQ-VBAKF algorithm that can adjust the PNCM and MNMCM simultaneously. This algorithm introduces the adaptive factor to self-tune the PNCM in real time based on the input signals. Furthermore, the MNMCM is estimated online using the VB approximation. According to the dynamic target tracking test results, the proposed AQ-VBAKF shows more accurate estimation and better robustness in comparison with other filtering methods. Moreover, for a general case in practical systems, when the initial PNCM and MNMCM are not appropriately set and deviate from the true values, the proposed algorithm has better self-adaptability and robustness than other algorithms. The proposed algorithm's computational burden is not heavy; thus, it can be applied in practical engineering applications.

For future prospects, the proposed filtering estimation algorithm can be applied in dynamic conditions, such as autonomous vehicles, target tracking, positioning, and so on. For example, autonomous vehicles inherently work under dynamic environments, and the environment noise statistics are unknown. The adaptability of the proposed AQ-VBAKF algorithm can adjust process and measurement noise covariances dynamically, which can significantly enhance the vehicle's perception accuracy and response. Our research provides a theoretical basis in terms of dynamic conditions to enhance the filtering estimation accuracy. In future work, the proposed algorithm will be employed in positioning systems to enhance positioning accuracy.

**Author Contributions:** Conceptualization, Y.C. and S.Z.; methodology, Y.C., X.W., and H.W.; software, Y.C. and H.W.; validation, Y.C., S.Z. and X.W.; investigation, Y.C., X.W. and H.W.; writing—original draft preparation, Y.C.; writing—review and editing, S.Z.; supervision, S.Z.; project administration, X.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available upon request from the corresponding author.

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

## References

1. Yuan, H.; Song, X. A Modified EKF for Vehicle State Estimation with Partial Missing Measurements. *IEEE Signal Process. Lett.* **2022**, *29*, 1594–1598.
2. Qiang, X.; Xue, R.; Zhu, Y. Two-Dimensional Monte Carlo Filter for a Non-Gaussian Environment. *Electronics* **2021**, *10*, 1385.
3. Yuan, T.; Zhao, R. LQR-MPC-Based Trajectory-Tracking Controller of Autonomous Vehicle Subject to Coupling Effects and Driving State Uncertainties. *Sensors* **2022**, *22*, 5556.
4. Cao, J.; Song, C.; Peng, S.; Song, S.; Zhang, X.; and Xiao, F. Trajectory Tracking Control Algorithm for Autonomous Vehicle Considering Cornering Characteristics. *IEEE Access* **2020**, *8*, 59470–59484.
5. López-Salcedo, J.A.; Peral-Rosado, J.A.; Seco-Granados, G. Survey on robust carrier tracking techniques. *IEEE Commun. Surv. Tuts.* **2014**, *16*, 670–688.
6. Xie, F.; Liang, G.; Chien, Y.-R. Highly Robust Adaptive Sliding Mode Trajectory Tracking Control of Autonomous Vehicles. *Sensors* **2023**, *23*, 3454.
7. Li, X.; Bar-Shalom, Y. A recursive multiple model approach to noise identification. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 671–684.
8. Jiang, B.; Sheng, W.; Zhang, R.; Han, Y.; Ma, X. Range tracking method based on adaptive “current” statistical model with velocity prediction. *Signal Process.* **2017**, *131*, 261–270.
9. Zha, F.; Guo, S.; Li, F. An improved nonlinear filter based on adaptive fading factor applied in alignment of SINS. *Optik* **2019**, *184*, 165–176.
10. Agamennoni, G.; Nieto, J.; Nebot, E.M. Approximate inference in state-space models with heavy-tailed noise. *IEEE Trans. Signal Process.* **2012**, *60*, 5024–5037.
11. Hou, B.; Wang, J.; He, Z.; Qin, Y.; Zhou, H.; Wang, D.; Li, D. Novel interacting multiple model filter for uncertain target tracking systems based on weighted Kullback–Leibler divergence. *J. Frankl. Inst.* **2020**, *357*, 13041–13084.
12. Sarkka, S.; Nummenmaa, A. Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Trans. Automat. Control* **2009**, *54*, 596–600.

13. Xu, D.; Wu, Z.; Huang, Y. A new adaptive Kalman filter with inaccurate noise statistics. *Circuits Syst. Signal Process.* **2019**, *38*, 4380–4404.
14. 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.
15. Ma, J.; Lan, H.; Wang, Z.; Wang, X.; Pan, Q.; Moran, B. Improved adaptive Kalman filter with unknown process noise covariance. In Proceedings of the 21st International Conference on Information Fusion, Cambridge, UK, 10–13 July 2018; pp. 2054–2058.
16. Xu, H.; Duan, K.; Yuan, H.; Xie, W.; Wang, Y. Black box variational inference to adaptive Kalman filter with unknown process noise covariance matrix. *Signal Process.* **2020**, *169*, 107413.
17. Huang, Y.; Zhu, F.; Jia, G.; Zhang, Y. A slide window variational adaptive Kalman filter. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2020**, *67*, 3552–3556.
18. Brown, R.G.; Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering*, 4th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2012; pp. 143–148.
19. Sarkka, S. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2013; pp. 56–62.
20. Tzikas, D.; Likas, A.; Galatsanos, N. The variational approximation for Bayesian inference. *IEEE Signal Process. Mag.* **2008**, *25*, 131–146.
21. Wang, G.; Wang, X.; Zhang, Y. Variational Bayesian IMM-Filter for JMSs With Unknown Noise Covariances. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1652–1661.
22. Chen, X.; Cao, L.; Cao, P.; Xiao, B. A higher-order robust correlation Kalman filter for satellite attitude estimation. *ISA Trans.* **2022**, *124*, 326–337.
23. Fang, J.; Yang, S. Study on innovation adaptive EKF for in-flight alignment of airborne POS. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 1378–1388.
24. Pan, C.; Gao, J.; Li, Z.; Qian, N.; Li, F. Multiple fading factors-based strong tracking variational Bayesian adaptive Kalman filter. *Measurement* **2021**, *176*, 109139.
25. Gao, W.; Li, J.; Zhou, G.; Li, Q. Adaptive Kalman Filtering with Recursive Noise Estimator for Integrated SINS/DVL Systems. *J. Navig.* **2015**, *68*, 142–161.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.