

Article

A Robust Adaptive Unscented Kalman Filter for Nonlinear Estimation with Uncertain Noise Covariance

Binqi Zheng ^{1,2}, Pengcheng Fu ^{1,2} , Baoqing Li ¹  and Xiaobing Yuan ^{1,*}

¹ Science and Technology on Microsystem Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China; bqzheng@mail.sim.ac.cn (B.Z.); fupc@mail.sim.ac.cn (P.F.); sinoiot@mail.sim.ac.cn (B.L.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: sinowsn@mail.sim.ac.cn; Tel.: +86-021-6907-5861

Received: 12 January 2018; Accepted: 5 March 2018; Published: 7 March 2018

Abstract: The Unscented Kalman filter (UKF) may suffer from performance degradation and even divergence while mismatch between the noise distribution assumed as a priori by users and the actual ones in a real nonlinear system. To resolve this problem, this paper proposes a robust adaptive UKF (RAUKF) to improve the accuracy and robustness of state estimation with uncertain noise covariance. More specifically, at each timestep, a standard UKF will be implemented first to obtain the state estimations using the new acquired measurement data. Then an online fault-detection mechanism is adopted to judge if it is necessary to update current noise covariance. If necessary, innovation-based method and residual-based method are used to calculate the estimations of current noise covariance of process and measurement, respectively. By utilizing a weighting factor, the filter will combine the last noise covariance matrices with the estimations as the new noise covariance matrices. Finally, the state estimations will be corrected according to the new noise covariance matrices and previous state estimations. Compared with the standard UKF and other adaptive UKF algorithms, RAUKF converges faster to the actual noise covariance and thus achieves a better performance in terms of robustness, accuracy, and computation for nonlinear estimation with uncertain noise covariance, which is demonstrated by the simulation results.

Keywords: Adaptive filter; data fusion; robust state estimation; nonlinear system; uncertain noise covariance

1. Introduction

Accurately and timely estimating the dynamic state in a nonlinear system is an important research area and has attracted considerable interest. For decades, many classic nonlinear estimation algorithms such as the extended Kalman filter (EKF) [1], unscented Kalman filter (UKF) [2], particle filter (PF) [3] and a large number of their varieties are put forward to address the estimation of dynamic state in nonlinear system. Among them, UKF is a considerably typical nonlinear filter algorithm, which has many merits such as simplicity in realization, high accuracy, and rapid convergence [4,5]. Consequently, it has been applied to many areas such as target tracking, navigation, energy estimation, structural dynamics, system identification, and vehicle positioning [6–11].

However, UKF as one of Kalman filter varieties is also the minimum mean square error (MMSE) estimator. It requires that the priori statistical characteristics of system noise be precisely known. Among these statistical characteristics, the process noise covariance and the measurement noise covariance are the most important as they directly regulate the impact of prediction values and measurements on the system state estimations [12,13]. However, in practical applications, it is difficult

to accurately describe covariances, which may suddenly change, leading to biased or even divergent filter solutions. Therefore, it is absolutely necessary to adaptively update the two noise covariances by using the information obtained in the filter process [14].

Recently, many efficient adaptive filter algorithms are proposed to address the above problem, which are usually divided into four categories: Bayesian, maximum likelihood (ML), correlation, and covariance-matching methods. Among the four categories, the covariance-matching method is the most computationally efficient [15]. Over the decade, many covariance-matching adaptive filter algorithms were proposed to address the above noise mismatch problem such as multiple model-based [7,16], innovation-based or residual-based [17–19] adaptive estimation algorithms. In [20], the authors propose an online adaptive UKF to update noise covariance matrices by minimizing a cost function built based on the errors between the covariance matrices of innovation and their corresponding estimations. However, considering a great amount of derivative calculations involved in the minimization process, this method is difficult to achieve the real-time performance. Shi et al. [21] adopt a modified Sage-Husa noise statistics estimator to estimate and adaptively adjust the process noise covariance, which is able to compensate the errors resulting from the change of noise statistics.

To obtain an accurate noise covariance, the Sage windowing method is frequently used to estimate system noise statistics based on windowing approximation [15]. For example, in [22], the authors present an adaptive UKF by combining the windowing and random weighting concepts and then extend the windowing method to nonlinear system. Reference [19] performs correction for the noise covariance depending on the type of the fault and also uses a windowing method when updating the process noise covariance. In [23], the authors propose an adaptive unscented particle filter which focuses on the adaptive estimation of the measurement noise covariance by using a windowing method with a fixed window size. Nevertheless, the sliding window method acquires the estimation of the current noise covariance by smoothing the relevant parameters during previous timesteps, which not only brings in heavy computational burden but also cannot adapt to the rapid changes of the system state.

Instead of using the windowing method, another approach to adjust noise covariance is to scale the noise covariance matrix with a time dependent variable. For example, in [24] a novel adaptive UKF algorithm is employed to estimate the attitudes of a pico satellite, in which the process noise covariance is adapted by a scaling factor calculated from the residual sequences. Furthermore, the authors propose a fault-detection method and the process noise covariance is only corrected when the fault occurs. In [25] the same authors put forward another adaptive UKF for the same purpose, where the measurement noise covariance is scaled by a scaling matrix instead of the process noise covariance, at the presence of measurement malfunction. Li et al. [26] presents a robust Masreliez–Martin UKF which can provide reliable state estimates in the presence of both unknown process noise and measurement noise covariance matrices. This method also adopts a fading factor to adjust the pre-designed process noise covariance. Nevertheless, the positive definite is not guaranteed in this scaling method [27]. Because the scaling factor is usually obtained by subtracting the two sum of traces of positive definite matrices. Thus, in some actual situations, the scaling factor may be negative, and then the new covariance matrix will not be a positive definiteness matrix. Additionally, the calculation of scale factors is also complex and unsuitable for devices with limited computational power, e.g., low-priced sensors.

In this paper, a robust adaptive UKF (RAUKF) is proposed for nonlinear state estimation with uncertain noise covariance to adaptively adjust the noise covariance according to its current estimation and previous value. Note that the uncertainty here contains two aspects: (i) the unknown of actual noise covariance in the estimation; and (ii) the dynamic change of actual noise covariance during the estimation. More specifically, at each timestep, a standard UKF will be implemented first to obtain current state estimations using the new acquired measurement. Then an online fault-detection mechanism is performed by using a statistical function to judge whether the current noise covariance needs to be updated. If needed, the innovation-based and residual-based methods are used to calculate

the estimations of process and measurement noise covariance, respectively. Furthermore, by utilizing a weighting factor, the filter obtains the new priori noise covariance values via combining the last ones with their current theoretical estimations. Finally, the state estimations will be corrected according to these new noise covariance matrices and previous state estimations. Unlike the above existing adaptive UKF algorithms based on the windowing or the scale factor, RAUKF need neither to store and smooth the relevant parameters during previous timesteps nor calculate scale factor at each timestep, which brings about a noteworthy reduction in the computational burden. Therefore, it is particularly well adapted for use in devices with limited computational power. Moreover, the proposed RAUKF algorithm could ensure that the new prior covariance matrices are certainly positive definite matrices, since it updates its covariance matrices with a weighting sum of two positive definite matrices. Simulation results show that RAUKF performs well in accuracy, robustness, and computation with uncertain noise covariance.

The rest of this paper is structured as follows. In Section 2, we introduce the nonlinear state estimation based on standard UKF (SUKF). Section 3 describes the proposed RAUKF algorithm. The simulation results of the proposed algorithm is investigated and evaluated in Section 4. Finally, Section 5 gives some conclusions to this paper.

2. Nonlinear State Estimation Based on SUKF

As we all know, the Kalman filter relies on two models, namely: a system or process model that describes the state of transition and it is used for prediction; and a measurement model that describes the relationship between state and measurements. In this paper, the state of system evolves according to the following discrete-time dynamic model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \quad (1)$$

where k is the sampling time index, $f(*)$ is the state transition function, \mathbf{x}_k is the state vector at timestep k , and \mathbf{w}_{k-1} is the process noise vector, assumed the zero-mean white Gaussian with covariance matrix \mathbf{Q}_{k-1} . The corresponding error covariance matrix of state \mathbf{x}_k is defined as

$$\mathbf{P}_k^{xx} = E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T], \quad (2)$$

where $\hat{\mathbf{x}}_{k|k}$ is the state estimation at timestep k , $E(*)$ is the expectation operation.

Subsequently, the measurements of system state at each timestep can be given indirectly by the following discrete observation model:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (3)$$

where $h(*)$ is the measurement function, \mathbf{z}_k is the measurement at timestep k , and \mathbf{v}_k is the measurement noise vector, which is also considered to be Gaussian with zero mean and covariance matrix \mathbf{R}_k .

With respect to the two system models, the nonlinear estimation based on SUKF can be briefly expressed as follows [2,6]. Note that the initial state vector $\hat{\mathbf{x}}_0$ and its error covariance matrix $\hat{\mathbf{P}}_0$ are assumed to be known and the initial noise covariance matrices \mathbf{Q}_0 and \mathbf{R}_1 are set to the positive definite matrices by users.

1. Compute weights with the initial parameter $0 < \omega_0 < 1$:

$$\omega_j = \frac{(1 - \omega_0)}{2n_x}, j = 1, \dots, 2n_x \quad (4)$$

$$\mathbf{c}^j = \sqrt{\frac{n_x}{1 - \omega_0}} \mathbf{r}^j, j = 1, \dots, 2n_x, \quad (5)$$

where n_x is the dimension of the state vector, $\{\mathbf{r}^j; j = 1, \dots, n_x\}$ is the j th standard basis vector and $\mathbf{r}^j = -\mathbf{r}^{(j-n_x)}$ when $j = n_x + 1, \dots, 2n_x$. How to select an appropriate ω_0 is also a meaningful topic and readers could refer to reference [2,28] for more details about the selection of ω_0 .

- At timestep k , establish symmetric sigma points $\boldsymbol{\phi}$ at timestep $k-1$ with the last state estimation $\hat{\mathbf{x}}_{k-1|k-1}$ and last estimation of state error covariance $\hat{\mathbf{P}}_{k-1|k-1}^{xx}$ which are both assumed to have been obtained from the previous step:

$$\mathbf{D}_{k-1|k-1} = (\hat{\mathbf{P}}_{k-1|k-1}^{xx})^{1/2} \quad (6)$$

$$\boldsymbol{\phi}_{k-1|k-1}^{(0)} = \hat{\mathbf{x}}_{k-1|k-1}, \quad \boldsymbol{\phi}_{k-1|k-1}^{(j)} = \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{D}_{k-1|k-1} \mathbf{c}^j, \quad j = 1, \dots, 2n_x. \quad (7)$$

- Predict the current states $\bar{\mathbf{x}}_{k|k-1}$ and its error covariance matrix $\bar{\mathbf{P}}_{k|k-1}^{xx}$:

$$\bar{\mathbf{x}}_{k|k-1} = \sum_{j=0}^{2n_x} \omega_j f(\boldsymbol{\phi}_{k-1|k-1}^{(j)}) \quad (8)$$

$$\bar{\mathbf{P}}_{k|k-1}^{xx} = \sum_{j=0}^{2n_x} \omega_j [f(\boldsymbol{\phi}_{k-1|k-1}^{(j)}) - \bar{\mathbf{x}}_{k|k-1}] \times [f(\boldsymbol{\phi}_{k-1|k-1}^{(j)}) - \bar{\mathbf{x}}_{k|k-1}]^T + \mathbf{Q}_{k-1}. \quad (9)$$

- Establish symmetric sigma points $\boldsymbol{\phi}$ about the state prediction:

$$\mathbf{D}_{k|k-1} = (\bar{\mathbf{P}}_{k|k-1}^{xx})^{1/2} \quad (10)$$

$$\boldsymbol{\phi}_{k|k-1}^{(0)} = \bar{\mathbf{x}}_{k|k-1}, \quad \boldsymbol{\phi}_{k|k-1}^{(j)} = \bar{\mathbf{x}}_{k|k-1} + \mathbf{D}_{k|k-1} \mathbf{c}^j, \quad j = 1, \dots, 2n_x. \quad (11)$$

- Define $\boldsymbol{\mu}_k = \mathbf{z}_k - h(\bar{\mathbf{x}}_{k|k-1})$ as the innovation vector at timestep k and the innovation covariance matrix is $\mathbf{P}_k^{zz} = E[\boldsymbol{\mu}_k * (\boldsymbol{\mu}_k)^T]$. Then, the predicted innovation covariance matrix $\bar{\mathbf{P}}_{k|k-1}^{zz}$ and cross covariance matrix $\bar{\mathbf{P}}_{k|k-1}^{xz}$ are separately determined by:

$$\bar{\mathbf{P}}_{k|k-1}^{zz} = \sum_{j=0}^{2n_x} \omega_j [h(\boldsymbol{\phi}_{k|k-1}^{(j)}) - \bar{\mathbf{z}}_{k|k-1}] \times [h(\boldsymbol{\phi}_{k|k-1}^{(j)}) - \bar{\mathbf{z}}_{k|k-1}]^T + \mathbf{R}_k \quad (12)$$

$$\bar{\mathbf{P}}_{k|k-1}^{xz} = \sum_{j=0}^{2n_x} \omega_j [f(\boldsymbol{\phi}_{k-1|k-1}^{(j)}) - \bar{\mathbf{x}}_{k|k-1}] \times [h(\boldsymbol{\phi}_{k|k-1}^{(j)}) - \bar{\mathbf{z}}_{k|k-1}]^T, \quad (13)$$

where $\bar{\mathbf{z}}_{k|k-1} = \sum_{j=0}^{2n_x} \omega_j h(\boldsymbol{\phi}_{k|k-1}^{(j)})$.

- Calculate Kalman gain \mathbf{K}_k and then obtain the estimation of current state $\hat{\mathbf{x}}_{k|k}$ and its error covariance matrix $\hat{\mathbf{P}}_{k|k}^{xx}$ with current actual measurement \mathbf{z}_k :

$$\mathbf{K}_k \triangleq \bar{\mathbf{P}}_{k|k-1}^{xz} (\bar{\mathbf{P}}_{k|k-1}^{zz})^{-1} \quad (14)$$

$$\hat{\mathbf{x}}_{k|k} = \bar{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \bar{\mathbf{z}}_{k|k-1}) \quad (15)$$

$$\hat{\mathbf{P}}_{k|k}^{xx} = \bar{\mathbf{P}}_{k|k-1}^{xx} - \mathbf{K}_k \bar{\mathbf{P}}_{k|k-1}^{zz} (\mathbf{K}_k)^T. \quad (16)$$

To run the SUKF, users need to provide \mathbf{Q}_{k-1} in Equation (9) and \mathbf{R}_k in Equation (12). Performance of a Kalman filter depends on how well users can configure the \mathbf{Q}_{k-1} and \mathbf{R}_k for current applications. Conventionally, they are often configured as constant matrices during the running of SUKF using a trial-and-error approach, which relies on the experience and background of users. Therefore, it is a challenge for users to configure initial \mathbf{Q}_0 and \mathbf{R}_1 at the beginning.

3. Robust Adaptive Unscented Kalman Filter

The SUKF algorithm works well under suitable \mathbf{Q} and \mathbf{R} . However, when users configure unsuitable noise covariance matrices as a priori or there is a fault either in system as a change in the noise covariance as a malfunction, SUKF may fail and thus its estimation results become inaccurate [19].

To address this challenge, we propose a robust adaptive unscented Kalman filter (RAUKF) algorithm. The algorithm adaptively adjusts \mathbf{Q} and \mathbf{R} based on the weighting combination of their current theoretical estimation values and the last data if a fault occurs in system. Note that SUKF works well in normal cases. Therefore, a fault-detection mechanism should be adopted to judge if it is necessary to revise noise covariance matrices. The adopted fault-detection mechanism in this letter is similar to that used in work [19] which uses the following statistical function to detect the fault:

$$\varphi_k = \boldsymbol{\mu}_k^T [\hat{\mathbf{P}}_{k|k-1}^{zz}]^{-1} \boldsymbol{\mu}_k, \quad (17)$$

The φ_k has a χ^2 distribution with s degree of freedom and s is the dimension of the $\boldsymbol{\mu}_k$. Suppose we wish to detect a fault with a reliability level of $1 - \sigma$, where σ is a chosen parameter. Then there is a threshold, $\chi_{\sigma,s}^2$, determined by the $\chi_{\sigma,s}^2$ distribution of φ_k such that

$$P(\varphi_k > \chi_{\sigma,s}^2) = \sigma \quad (18)$$

Thus, using the preselected value of σ , we define a corresponding fault detection threshold $\chi_{\sigma,s}^2$ from Equation (18), such that a system fault can be detected with a reliability level of $1 - \sigma$.

3.1. Adaptive Adjustment of \mathbf{Q}

The innovation sequence $\boldsymbol{\mu}_k$ represents the additional information available to the filter as a consequence of the incoming new measurements. Hence, it is considered as the most relevant information for the filter adaptation and can be used to estimate the noise covariance [23]. According to Equation (1), the process noise can be represented as $\mathbf{w}_{k-1} = \mathbf{x}_k - f(\mathbf{x}_{k-1})$.

Furthermore, from equations in SUKF, it yields

$$\begin{aligned} \hat{\mathbf{w}}_{k-1} &= \hat{\mathbf{x}}_k - f(\hat{\mathbf{x}}_{k-1|k-1}) \approx \hat{\mathbf{x}}_k - \bar{\mathbf{x}}_{k|k-1} \\ &= \mathbf{K}_k(\mathbf{z}_k - \bar{\mathbf{z}}_{k|k-1}) \approx \mathbf{K}_k \boldsymbol{\mu}_k. \end{aligned} \quad (19)$$

Note that there are two approximations via linearization used in the above derivation, $\bar{\mathbf{x}}_{k|k-1} \approx f(\hat{\mathbf{x}}_{k-1|k-1})$ and $\bar{\mathbf{z}}_{k|k-1} \approx h(\bar{\mathbf{x}}_{k|k-1})$, in which the forward evolution of the mean at time $k - 1$ is approximated by the mean computed from the forward evolved sigma points; respectively the observation of the mean is approximated by the mean observation, computed via the observations of the sigma points.

Therefore, the estimation of process noise covariance matrix \mathbf{Q}_{k-1} can be estimated as:

$$\mathbf{Q}_{k-1} = cov(\hat{\mathbf{w}}_{k-1}) = \mathbf{K}_k E[\boldsymbol{\mu}_k \boldsymbol{\mu}_k^T] \mathbf{K}_k^T, \quad (20)$$

where $cov(*)$ is the covariance operation and $E(*)$ is the expectation operation.

To implement above equation, $E(\boldsymbol{\mu}_k \boldsymbol{\mu}_k^T)$ usually approximated by means of averaging $\boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$ over time using a windowing method. Instead of using moving window methods (like in works [15,19,23,29]), this paper adaptively adjusts \mathbf{Q} by utilizing a weighting factor λ to balance the last noise covariance value and current estimation. The weighting factor λ is set with a lower limit $\lambda_0 \in (0, 1)$ to ensure the update strength and will increase as the rise of the φ_k if the φ_k exceeds a preset threshold. Therefore, the system process noise covariance matrix is updated as:

$$\mathbf{Q}_{k-1} = (1 - \lambda) \mathbf{Q}_{k-1} + \lambda (\mathbf{K}_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \mathbf{K}_k^T), \quad (21)$$

$$\lambda = \max\{\lambda_0, (\varphi_k - a \times \chi_{\sigma,s}^2) / \varphi_k\}, \quad (22)$$

where a ($a > 0$) is a tuning parameter dependent on the actual environment, and a larger a implies a higher probability of using the default λ_0 . Specially, how the trade-off between λ_0 and a will determine how sensitive the covariance update is to the new innovation statistics.

3.2. Adaptive Adjustment of \mathbf{R}

The measurement noise covariance matrix \mathbf{R} at timestep k can also be estimated by the innovation-based approach as follows:

$$\hat{\mathbf{R}}_k = E[\boldsymbol{\mu}_k \boldsymbol{\mu}_k^T] - \bar{\mathbf{S}}_{k|k-1}^{zz}, \quad (23)$$

where $\bar{\mathbf{S}}_{k|k-1}^{zz} = \sum_{j=0}^{2n_x} \omega_j [h(\boldsymbol{\phi}_{k|k-1}^{(j)}) - \bar{\mathbf{z}}_{k|k-1}] \times [h(\boldsymbol{\phi}_{k|k-1}^{(j)}) - \bar{\mathbf{z}}_{k|k-1}]^T$. Readers could refer to reference [30] for more details about the derivation process. Generally speaking, the \mathbf{R}_k should be positive definite as it is a covariance matrix. However, its estimation $\hat{\mathbf{R}}_k$ from Equation (23) cannot guarantee to be a positive definite matrix, since it is obtained by subtracting the two positive definite matrix. Then, if there are negative parameters in the main diagonal of noise covariance matrix, the adaptive filter will suddenly diverge [22]. To obtain a positive definite matrix $\hat{\mathbf{R}}_k$, a residual-based approach is used. From Equation (3) the measurement noise at timestep k can be derived as $\mathbf{v}_k = \mathbf{z}_k - h(\mathbf{x}_k)$. Then, the residual vector at timestep k is given by

$$\boldsymbol{\varepsilon}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k}). \quad (24)$$

Furthermore, according to work [23], the estimation of measurement noise covariance based on the residual vector $\boldsymbol{\varepsilon}_k$ is given by

$$\hat{\mathbf{R}}_k = \text{cov}(\hat{\mathbf{v}}_k) = E[\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T] + \hat{\mathbf{S}}_{k|k}^{zz}, \quad (25)$$

where

$$\hat{\mathbf{S}}_{k|k}^{zz} = \sum_{j=0}^{2n_x} \omega_j [h(\boldsymbol{\phi}_{k|k}^{(j)}) - \hat{\mathbf{z}}_{k|k}] \times [h(\boldsymbol{\phi}_{k|k}^{(j)}) - \hat{\mathbf{z}}_{k|k}]^T, \quad (26)$$

$$\boldsymbol{\phi}_{k|k}^{(0)} = \hat{\mathbf{x}}_{k|k}, \quad \boldsymbol{\phi}_{k|k}^{(j)} = \hat{\mathbf{x}}_{k|k} + \mathbf{D}_{k|k} \mathbf{c}^{(j)}, \quad j = 1, \dots, 2n_x, \quad \mathbf{D}_{k|k} = (\hat{\mathbf{P}}_{k|k}^{xx})^{1/2}, \quad (27)$$

$$\hat{\mathbf{z}}_{k|k} = \sum_{j=0}^{2n_x} \omega_j h(\boldsymbol{\phi}_{k|k}^{(j)}). \quad (28)$$

Similar to the method used in the previous section, the \mathbf{R}_k is updated utilizing a weighting factor δ set with a lower limit $\delta_0 \in (0, 1)$ as:

$$\mathbf{R}_k = (1 - \delta) \mathbf{R}_k + \delta [\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T + \hat{\mathbf{S}}_{k|k}^{zz}] \quad (29)$$

$$\delta = \max\{\delta_0, (\varphi_k - b \times \chi_{\sigma,s}^2) / \varphi_k\}, \quad (30)$$

where b ($b > 0$) is also a tuning parameter dependent on the actual environment, and the selection of the b and δ_0 is the same as that of the a and λ_0 .

3.3. Correct Estimations

Once the covariance matrices \mathbf{Q}_{k-1} and \mathbf{R}_k are updated, current state estimations should be corrected with the new \mathbf{R}_k , \mathbf{Q}_{k-1} and the estimations $\hat{\mathbf{x}}_{k|k}$, $\hat{\mathbf{P}}_{k|k}^{xx}$ to obtain more accurate state estimations. The correction process is described as follows:

1. Compute the new predicted error covariance matrix $\bar{\mathbf{P}}_{k|k}^{xx}$ and cross covariance matrix $\bar{\mathbf{P}}_{k|k}^{xz}$, which treats the acquired state $\hat{\mathbf{x}}_{k|k}$ as the predicted state of timestep k :

$$\bar{\mathbf{P}}_{k|k}^{xx} = \sum_{j=0}^{2n_x} \omega_j [\boldsymbol{\phi}_{k|k}^{(j)} - \hat{\mathbf{x}}_{k|k}]^T \times [\boldsymbol{\phi}_{k|k}^{(j)} - \hat{\mathbf{x}}_{k|k}] + \mathbf{Q}_{k-1}, \quad (31)$$

$$\bar{\mathbf{P}}_{k|k}^{xz} = \sum_{j=0}^{2n_x} \omega_j [\boldsymbol{\phi}_{k|k}^{(j)} - \hat{\mathbf{x}}_{k|k}] \times [h(\boldsymbol{\phi}_{k|k}^{(j)}) - \hat{\mathbf{z}}_{k|k}]^T \quad (32)$$

2. Evaluate the updated innovation covariance matrix $\hat{\mathbf{P}}_{k|k}^{zz}$ and Kalman gain $\hat{\mathbf{K}}_k$:

$$\hat{\mathbf{P}}_{k|k}^{zz} = \hat{\mathbf{S}}_{k|k}^{zz} + \mathbf{R}_k, \quad \hat{\mathbf{K}}_k \triangleq \bar{\mathbf{P}}_{k|k}^{xz} (\hat{\mathbf{P}}_{k|k}^{zz})^{-1} \quad (33)$$

3. Correct the estimation of current state $\hat{\mathbf{x}}_{k|k}$ and its error covariance matrix $\hat{\mathbf{P}}_{k|k}^{xx}$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k} + \hat{\mathbf{K}}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k}) \quad (34)$$

$$\hat{\mathbf{P}}_{k|k}^{xx} = \bar{\mathbf{P}}_{k|k}^{xx} - \hat{\mathbf{K}}_k \bar{\mathbf{P}}_{k|k}^{xz} (\hat{\mathbf{K}}_k)^T. \quad (35)$$

In this subsection, we use the estimation $\hat{\mathbf{x}}_{k|k}$ and its sigma point $\boldsymbol{\phi}_{k|k}^{(j)}$ to replace the previous forecast $\bar{\mathbf{x}}_{k|k-1}$ and its sigma point $\boldsymbol{\phi}_{k|k-1}^{(j)}$ as the new input values in the correction phase. Although this method will incorporate the previous mis-estimated \mathbf{Q}_{k-1} and \mathbf{R}_k into the final estimations, the previous estimation $\hat{\mathbf{x}}_{k|k}$ obtained from Equation (15) is more precise than the prediction value $\bar{\mathbf{x}}_{k|k-1}$. Thus, in this paper we adopt the estimation $\hat{\mathbf{x}}_{k|k}$ and its sigma point $\boldsymbol{\phi}_{k|k}^{(j)}$ as the input values of our correction phase, which is like in many iterated UKF algorithms, e.g., [31,32].

From Equation (21) and Equation (29), we can find that the proposed RAUKF adopts a one-step update instead of using the windowing procedure, which brings about the decrease of computation burden and is friendly to the sensors with limited computation power and storage capacity. Moreover, the new noise covariance matrices obtained via weighting sum of previous values and current estimation are certainly the positive definite matrices. The proof is given as follows. The overall procedure of the proposed RAUKF algorithm is summarized in Algorithm 1.

Theorem 1. *The new obtained prior noise covariance matrices in Equations (21) and Equation (29) are both the positive definite matrices.*

Proof. At the beginning of the proposed algorithm, the initial noise covariance matrices \mathbf{Q}_0 and \mathbf{R}_1 is set to the positive definite matrices by users. At timestep k , the prior noise covariance matrices will be updated if the system fault occurs. The new noise covariance matrices are obtained via weighting sum of their previous values and current estimation which are described in Equations (21) and Equation (29). In the equations, the previous noise covariance matrices \mathbf{Q}_{k-1} and \mathbf{R}_k are already both the positive definite matrices. The expression $\mathbf{K}_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \mathbf{K}_k^T$ in Equation (21) is a positive definite matrix as it is the product of $(\mathbf{K}_k \boldsymbol{\mu}_k)$ and its transpose $(\mathbf{K}_k \boldsymbol{\mu}_k)^T$. Similarly, the expression $\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T$ in Equation (29) is also a positive definite matrix. Furthermore, from Equation 26, the new obtained $\hat{\mathbf{S}}_{k|k}^{zz}$ is a weighting sum of products which are all positive definite matrices. Thus, it is also a positive definite matrix. Additionally, the weight factors $\lambda, \delta \in (0, 1)$. Therefore, the new obtained prior noise covariance matrices \mathbf{Q}_{k-1} and \mathbf{R}_k in Equations (21) and (29) are both the positive definite matrices. \square

Algorithm 1. The robust adaptive unscented Kalman filter (RAUKF) algorithm.

Input: $f(\cdot), h(\cdot), \hat{\mathbf{x}}_0, \mathbf{Q}_0, \mathbf{R}_1, \hat{\mathbf{P}}_0, \lambda_0, \delta_0, \omega_0, \chi_{\sigma, s}^2$.

- 1: Initialization:
- 2: $\omega_j = (1 - \omega_0)/2n_x, \mathbf{c}^0 = \sqrt{\frac{n_x}{1-\omega_0}}, \mathbf{c}^{(j)} = \sqrt{\frac{n_x}{1-\omega_0}} \mathbf{r}^{(j)}, j = 1, \dots, 2n_x$.
- 3: **for** $k = 1 \rightarrow K$ **do**
- 4: Implement SUKF to obtain $\hat{\mathbf{x}}_{k|k}, \hat{\mathbf{P}}_{k|k-1}^{zz}, \mathbf{K}_k, \hat{\mathbf{P}}_{k|k}^{xx}$.
- 5: Perform the fault-detection mechanism:
- 6: $\varphi_k = \boldsymbol{\mu}_k^T [\hat{\mathbf{P}}_{k|k-1}^{zz} + \mathbf{R}_k]^{-1} \boldsymbol{\mu}_k$
- 7: **if** $\varphi_k > \chi_{\sigma, s}^2$, **then**
- 8: Update the \mathbf{Q}_{k-1} and \mathbf{R}_k :
- 9: $\boldsymbol{\mu}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}), \boldsymbol{\varepsilon}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k});$
- 10: $\hat{\mathbf{S}}_{k|k}^{zz} = \sum_{j=0}^{2n_x} \omega_j [h(\boldsymbol{\phi}_{k|k}^{(j)}) - \bar{\mathbf{z}}_{k|k}] \times [h(\boldsymbol{\phi}_{k|k}^{(j)}) - \bar{\mathbf{z}}_{k|k}]^T;$
- 11: $\delta = \max\{\delta_0, (\varphi_k - a \times \chi_{\sigma, s}^2) / \varphi_k\}; \lambda = \max\{\lambda_0, (\varphi_k - b \times \chi_{\sigma, s}^2) / \varphi_k\};$
- 12: $\mathbf{Q}_{k-1} \leftarrow (1 - \lambda)\mathbf{Q}_{k-1} + \lambda(\mathbf{K}_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \mathbf{K}_k^T); \mathbf{R}_k \leftarrow (1 - \delta)\mathbf{R}_k + \delta[\boldsymbol{\varepsilon}_k(\boldsymbol{\varepsilon}_k)^T + \hat{\mathbf{S}}_{k|k}^{zz}].$
- 13: Correct state estimations:
- 14: $\hat{\mathbf{K}}_k \triangleq \hat{\mathbf{P}}_{k|k}^{xz} (\hat{\mathbf{P}}_{k|k}^{zz})^{-1};$
- 15: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \hat{\mathbf{K}}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k}); \hat{\mathbf{P}}_{k|k}^{xx} = \hat{\mathbf{P}}_{k|k-1}^{xx} - \hat{\mathbf{K}}_k \hat{\mathbf{P}}_{k|k}^{xz} (\mathbf{K}_k)^T;$
- 16: **end if**
- 17: $\mathbf{Q}_k \leftarrow \mathbf{Q}_{k-1}, \mathbf{R}_{k+1} \leftarrow \mathbf{R}_k$.
- 18: Save the $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{P}}_{k|k}^{xx}$.
- 19: **end for**

4. Simulation Results

To highlight the merits of RAUKF, we conduct the simulation experiment on a simple vehicle-tracking problem in two different cases. In the first experiment, the actual noise covariance matrices are assumed unknown and impossible to estimate off-line. In the second experiment, the actual noise covariance of the system will dynamically change during the estimation. The experiment results are compared with that of SUKF algorithm with fixed noise covariance, AUKF algorithm with a moving window method (called AUKF-window, the window size is set to 10 here) [23], and AUKF algorithm with scaling factor method (called AUKF-scaling-factor) [24]. Note that the AUKF-scaling-factor method in [24] only updates its process noise covariance matrix when the fault occurs. Thus, this method will only be compared with the proposed RAUKF in the second case. All simulations are executed on an Intel Core i5-2520M CPU 2.50 GHz \times 4 personal computer with 10 GB of random-access memory.

$\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ denotes the vehicle state at discrete timestep k , where $p_k = (x_k, y_k)$ is the current vehicle position and (\dot{x}_k, \dot{y}_k) is its velocity. Δ is the constant sampling time interval between two successive measurements. Meanwhile, a stationary radar is assumed to be used to track the vehicle at position $(0, 0)$. Its measurement vector is described by $\mathbf{z}_k = [\rho_k, \theta_k, v_k]$, where ρ_k and θ_k are respectively the distance and angle between the radar and vehicle, v_k is the vehicle speed. Therefore, the system models are given by

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan(y_k/x_k) \\ \sqrt{\dot{x}_k^2 + \dot{y}_k^2} \end{bmatrix} + \mathbf{v}_k, \end{cases} \quad (36)$$

where \mathbf{A} is the state transition matrix.

In this paper, the root mean-squared error in position at each timestep, $RMSE_p$, and its average, $ARMSE_p$, are adopted as the indications of tracking accuracy, since it yields a combined measurement of the bias and variance of a filter estimate [33,34]. The $ARMSE_p$ is given by

$$ARMSE_p = \sqrt{\frac{1}{N_m K} \sum_{i=1}^{N_m} \sum_{k=1}^K [(\hat{x}_k^{(i)} - x_k)^2 + (\hat{y}_k^{(i)} - y_k)^2]}, \quad (37)$$

and the $RMSE_p$ at timestep k yields

$$RMSE_p(k) = \sqrt{\frac{1}{N_m} \sum_{i=1}^{N_m} [(\hat{x}_k^{(i)} - x_k)^2 + (\hat{y}_k^{(i)} - y_k)^2]}, \quad (38)$$

where $(\hat{x}_k^{(i)}, \hat{y}_k^{(i)})$ is the estimated vehicle position in timestep k at i -th Monte Carlo run. N_m is the number of Monte Carlo runs and $K = 100$ is the number of time sample in one run. $\chi_{\sigma, s}^2$ is taken as 2.37, and this value comes from chi-square distribution when the degree of freedom is 3 and the reliability level is 50%. Other parameters in the simulations are set as follows:

$$\mathbf{Q}^\circ = 9 * \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}^\circ = \text{diag}([1, 0.0001, 9]), \quad \hat{\mathbf{P}}_0 = \text{diag}([2, 3, 2, 3]),$$

$$\hat{\mathbf{x}}_0 = [0, 10, 0, 10], \quad \Delta = 0.1,$$

$$N_m = 1000, \quad \lambda_0 = \delta_0 = 0.2,$$

where \mathbf{Q}° and \mathbf{R}° are the actual process and measurement noise covariance matrices of our simulation system, respectively.

4.1. Selection of Tuning Parameters a and b

Before implementing the simulation experiments, we need to select suitable tuning parameter a and b . According to Equations (21), (22) and (29), (30), the adaptive adjustments of Q_{k-1} and R_k have the same theories and structures. Thus, in this paper, the parameter a and b are given the same values.

To select suitable values for the two tuning parameters, they are varied from 1 to 8 in two different situations. In first situation, the initial prior noise covariance matrices are set to the actual noise covariance matrices, namely, $Q_0 = Q^\circ, R_1 = R^\circ$. While, the initial prior noise covariance matrices in another one is that $Q_0 = 100 \times Q^\circ, R_1 = 0.01 \times R^\circ$.

As shown in Figure 1a, the improvement of estimation accuracy diminishes and seems to be trivial after $a > 4$. In this situation, the prior noise covariance is close to the actual one. Thus, the weighting factor λ should be as small as possible and then a large a implies that the λ will be set to its lower limit λ_0 with a very high probability. From Figure 1b, the $ARMSE_p$ decreases first and then rising a little. In this situation, a moderate tuning parameter is necessary, as too large or too small tuning parameter may have negative influences on the covariance-matching. The principle of selecting δ is the same with that of λ . Therefore, in the following simulation experiments, a and b are chosen as 5.

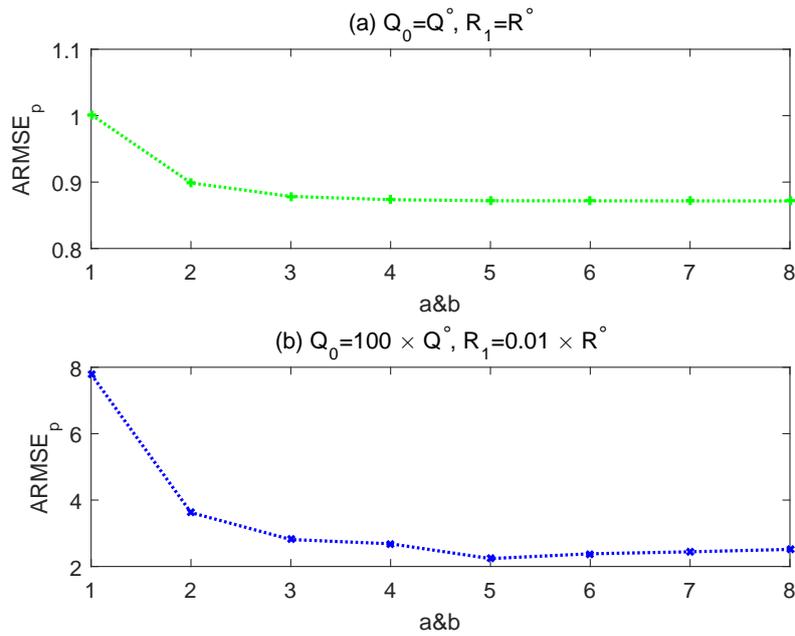


Figure 1. $ARMSE_p$ results of different weighting parameters in two different situations.

4.2. Simulation Results in Unknown Noise Environment

In the first simulation case, we evaluate the performance of RAUKF with arbitrary initial noise covariance matrices. Specifically, Q_0 and R_1 are varied apart from small to large by scaling Q^0 and R^0 to test the accuracy and robustness of RAUKF. Under the same setup of simulations, the $ARMSE_p$ results of different algorithms are respectively summarized from Tables 1–3.

Comparing the three tables, one can observe that in general the proposed RAUKF achieves the best performance in robustness and accuracy among the three algorithms. In detail, from Table 1, the $ARMSE_p$ results of SUKF are relatively greater when the Q_0 or R_1 is set to a large or small value than those of a normal range, especially in the situation that the initial noise covariance matrices are considerably lower than the actual ones. In many cases, the $ARMSE_p$ results are more robust in AUKF-window rather than in SUKF by contrasting Tables 1 and 2. However, owing to the window smoothing operations, the $ARMSE_p$ results of AUKF-window are almost similar in most of different initial noise covariance matrices. Furthermore, they are slightly above that of SUKF except at the bottom left of the table. It says that the windowing-based AUKF improves the robustness of estimation but reduces its accuracy. Meanwhile, in some cases (e.g., $m = 0.01$, $n = 100$) the $ARMSE_p$ results are also large. As shown in Table 3, the performance improvement of RAUKF is more obvious when the $ARMSE_p$ results of AUKF-window and SUKF are both large. For example, Figure 2 shows the $RMSE_p$ results of the three algorithms at each timestep when $Q_0 = 100 \times Q^0$ and $R_1 = 0.01 \times R^0$. Additionally, in the normal situations, the $ARMSE_p$ results of RAUKF are also almost the same as those of SUKF. In summarize, RAUKF achieves a good performance even if users know very little about the actual noise covariance and uses an unsuitable approximation.

As for the computational performance, all algorithms are run in the same computer and their running time results are all averaged over 1000 runs. The time consumption of RAUKF is 28.40 ms for each run, which is slightly higher than that of SUKF (23.30 ms). This is because RAUKF needs to update the noise covariance matrices and correct the state estimations. However, AUKF-window's computation time is 52.04 ms, which is much greater than that of SUKF and RAUKF. This is due to it involves a vast number of calculations in the updates of noise covariance utilizing a windowing method. Furthermore, the length of the window in AUKF-window determines the amount of the measurements that need to be accumulated and averaged to calculate the estimated covariance matrices [23]. Thus, the longer the length of the window is, the higher the computation is.

Table 1. $ARMSE_p$ results of SUKF.

$ARMSE_p$		$R_1 = m \times R^\circ$				
		$m = 0.01$	$m = 0.1$	$m = 1$	$m = 10$	$m = 100$
$Q_0 = n \times Q^\circ$	$n = 0.01$	7.1569	5.0975	1.0752	1.2882	1.4639
	$n = 0.1$	7.1716	5.0995	0.8666	1.0588	1.2598
	$n = 1$	7.1879	5.1180	0.8563	0.8496	1.0433
	$n = 10$	7.1974	5.0640	0.9520	0.8692	0.9305
	$n = 100$	21.3282	18.1290	1.0739	1.3955	2.7678

Table 2. $ARMSE_p$ results of AUKF-window (window size = 10).

$ARMSE_p$		$R_1 = m \times R^\circ$				
		$m = 0.01$	$m = 0.1$	$m = 1$	$m = 10$	$m = 100$
$Q_0 = n \times Q^\circ$	$n = 0.01$	1.6672	1.6577	1.6136	1.5858	1.5636
	$n = 0.1$	1.6558	1.6561	1.6128	1.5851	1.5534
	$n = 1$	1.6629	1.6470	1.6054	1.5791	1.5613
	$n = 10$	5.0705	1.6141	1.5831	1.5578	1.5492
	$n = 100$	7.2723	6.6321	1.6130	1.5886	1.5847

Table 3. $ARMSE_p$ results of RAUKF.

$ARMSE_p$		$R_1 = m \times R^\circ$				
		$m = 0.01$	$m = 0.1$	$m = 1$	$m = 10$	$m = 100$
$Q_0 = n \times Q^\circ$	$n = 0.01$	1.5843	1.4737	1.1664	1.2476	1.4639
	$n = 0.1$	1.4917	1.3922	1.1271	1.0368	1.2598
	$n = 1$	1.3424	1.2940	1.0827	0.8472	1.0433
	$n = 10$	1.8842	1.6269	1.1120	0.8752	0.9305
	$n = 100$	2.4231	3.7601	1.5362	1.3892	2.5616

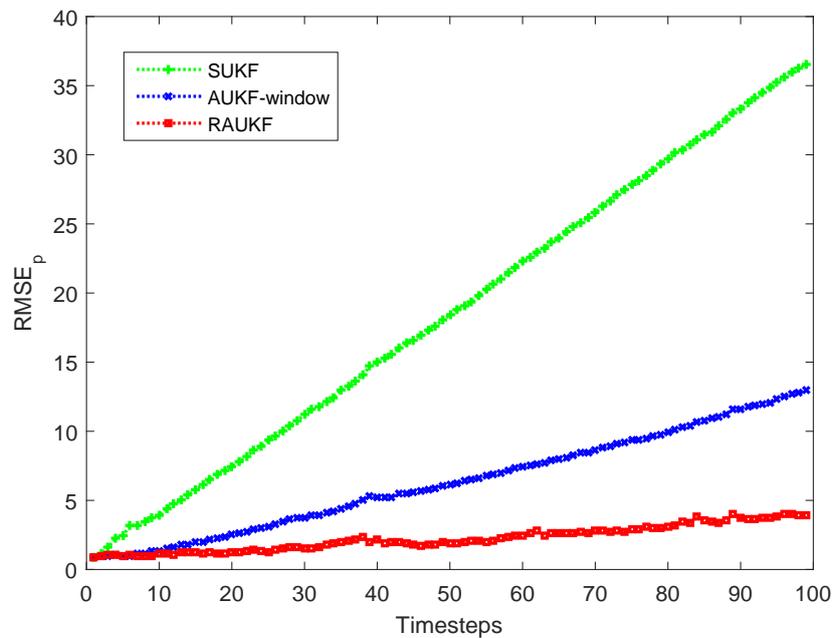


Figure 2. Comparison of $RMSE_p(k)$ results of different algorithms when $Q_0 = 100 \times Q^\circ$ and $R_1 = 0.01 \times R^\circ$.

4.3. Simulation Results in Dynamic Noise Environment

In the second case, we assume that the actual noise intensity of the vehicle occurs a sudden dramatic change during the tracking to highlight the robustness of RAUKF. Therefore, the actual process noise covariance in this simulation is set as

$$\begin{cases} \mathbf{Q}_k^{(v)} = \mathbf{Q}^\circ, k = 1, \dots, 20 \\ \mathbf{Q}_k^{(v)} = 100 \times \mathbf{Q}^\circ, k = 21, \dots, K. \end{cases} \quad (39)$$

It should be noted that the parameter δ is set as 0, since the $\mathbf{R}_k^{(v)}$ of the vehicle do not change during the simulation. In this case, the proposed RAUKF algorithm will be compared with AUKF-window, AUKF-scaling-factor, and SUKF for a comprehensive evaluation. Meanwhile, the \mathbf{R}_k of AUKF-window keeps fixed for fairness to compare with RAUKF. $\mathbf{R}_1 = \mathbf{R}^\circ$ and other parameters are the same with previous setup.

Figure 3 shows one of comparison results of different algorithms in state estimation at each timestep and Figure 4 presents the $RMSE_p$ results of the four algorithms at each timestep, from which we can see that RAUKF successfully track the vehicle after an abrupt change of the $\mathbf{Q}_k^{(v)}$. Moreover, SUKF cannot track the vehicle well now due to the fixed pseudonoise intensity, which couldn't provide enough drive power to the related estimation. Thus, its $RMSE_p$ results increase rapidly after timestep 20. As for the comparison of AUKF-window, AUKF-scaling-factor, and RAUKF, their \mathbf{Q}_k will increase after timestep 20 because of the adaptive adjusting mechanisms. Thus, the $RMSE_p$ results of them are all steady decline after timestep 20. However, the proposed RAUKF performs better than that of AUKF-window method and AUKF-scaling-factor method.

More concretely, as shown in Table 4, the averaged $RMSE_p$ results of the four different algorithms before timestep 20 are almost the same, which means that all of them could deal with the normal situation well. Then, the actual process noise of system changes at the timestep 20, the averaged $RMSE_p$ results of the four different algorithms begin to be different. There is no doubt that SUKF algorithm performs the worst among the four algorithms after timestep 20 as the mismatch between its prior noise covariance and the actual one. The averaged $RMSE_p$ of the proposed RAUKF after timestep 20 is the lowest compared with that of AUKF-window method and AUKF-scaling-factor method, which shows the superiority of RAUKF in dealing with the nonlinear estimation in dynamic noise environment. In detail, the convergence speed of $RMSE_p$ of RAUKF is the fastest among the three different adaptive algorithms (about 10 sample steps).

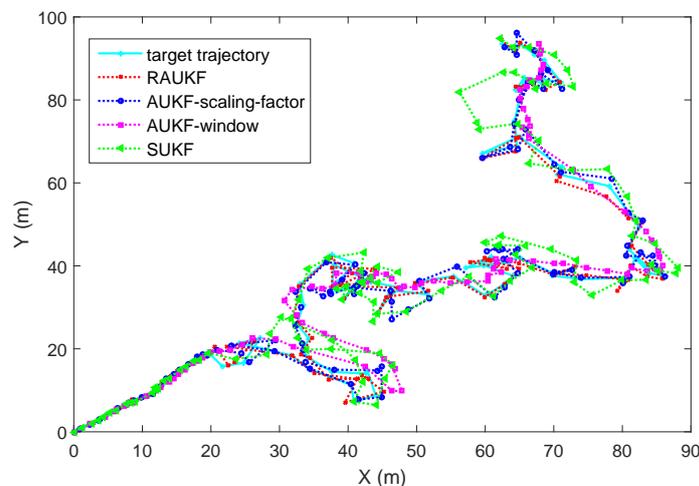


Figure 3. Comparison of state estimation results of different algorithms before and after an abrupt change of the $\mathbf{Q}_k^{(v)}$.

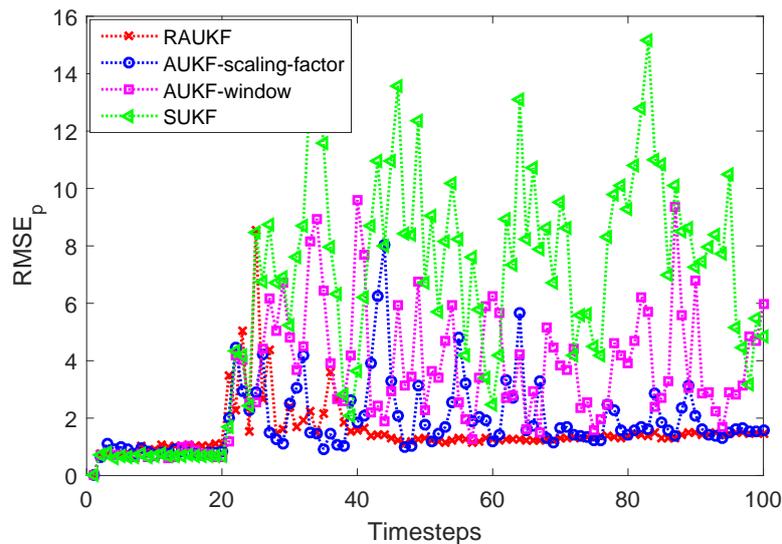


Figure 4. Comparison of $RMSE_p(k)$ results of different algorithms before and after an abrupt change of the $Q_k^{(v)}$.

Table 4. Averaged $RMSE_p$ before and after timestep 20.

Averaged $RMSE_p$	SUKF	AUKF-Window	AUKF-Scaling-Factor	RAUKF
Before timestep 20	0.6886	0.7397	0.7970	0.8163
After timestep 20	7.6859	4.0566	2.1757	1.6093

5. Conclusions

In this paper, a robust adaptive unscented Kalman filter (RAUKF) is proposed for the state estimation of nonlinear system under a sequential framework with uncertain noise covariance. A fault-detection mechanism is used to judge if it is necessary to update the noise covariance matrices. If necessary, RAUKF will adjust Q and R based on the weighting combination of current theoretical estimation and previous values. The state estimations will then be corrected. Unlike many existing adaptive UKF algorithms based on the windowing or the scale factor, RAUKF need neither to store and smooth the relevant parameters during previous timesteps nor calculate scale factor at each timestep, which brings about a noteworthy reduction in the computational burden. Therefore, it is particularly well adapted for use in devices with limited computational power. Moreover, the proposed RAUKF algorithm could ensure that the new prior covariance matrices are certainly positive definite matrices, since it updates its covariance matrices with a weighting sum of two positive definite matrices. Simulations results from two aspects demonstrate that the proposed RAUKF performs well in accuracy and robustness on nonlinear estimation with uncertain noise covariance.

This work adaptively updates the process and measurement noise covariance simultaneously when the system fault is detected. Thus, in the further research work, we will focus on investigating how to exactly detect the type of the fault (either a process noise uncertainty or a measurement malfunction). Then the process noise covariance or measurement noise covariance will be revised respectively according to the fault type.

Author Contributions: The work presented here was carried out in collaboration between all authors. Binqi Zheng and Xiaobing Yuan conceived and designed the experiments; Pengcheng Fu performed the experiments; Baoqing Li analyzed the data and contributed analysis tools; Binqi Zheng wrote the paper.

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

References

1. Reif, K.; Unbehauen, R. The extended Kalman filter as an exponential observer for nonlinear systems. *IEEE Trans. Signal Process.* **1999**, *47*, 2324–2328.
2. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422.
3. Carpenter, J.; Clifford, P.; Fearnhead, P. Improved particle filter for nonlinear problems. *IEE Proc. Radar Sonar Navig.* **2002**, *146*, 2–7.
4. Lee, D.J. Nonlinear Estimation and Multiple Sensor Fusion Using Unscented Information Filtering. *IEEE Signal Process. Lett.* **2008**, *15*, 861–864.
5. Gustafsson, F.; Hendeby, G. Some Relations Between Extended and Unscented Kalman Filters. *IEEE Trans. Signal Process.* **2012**, *60*, 545–555.
6. Fu, P.; Tang, H.; Cheng, Y.; Li, B.; Qian, H.; Yuan, X. An energy-balanced multi-sensor scheduling scheme for collaborative target tracking in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*.
7. Xu, Q.; Li, X.; Chan, C.Y. A Cost-Effective Vehicle Localization Solution Using an Interacting Multiple Model Unscented Kalman Filters (IMM-UKF) Algorithm and Grey Neural Network. *Sensors* **2017**, *17*, 1431.
8. Zhang, P.; Gu, J.; Milios, E.E.; Huynh, P. Navigation with IMU/GPS/digital compass with unscented Kalman filter. In Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; Volume 29, pp. 1497–1502.
9. Azam, S.E.; Ghisi, A.; Mariani, S. Parallelized sigma-point Kalman filtering for structural dynamics. *Comput. Struct.* **2012**, *92–93*, 193–205.
10. Azam, S.E.; Bagherinia, M.; Mariani, S. Stochastic system identification via particle and sigma-point Kalman filtering. *Sci. Iranica* **2012**, *19*, 982–991.
11. Vaccarella, A.; Momi, E.D.; Enquobahrie, A.; Ferrigno, G. Unscented Kalman Filter Based Sensor Fusion for Robust Optical and Electromagnetic Tracking in Surgical Navigation. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 2067–2081.
12. Ardeshiri, T.; Ozkan, E.; Orguner, U.; Gustafsson, F. Approximate Bayesian Smoothing with Unknown Process and Measurement Noise Covariances. *IEEE Signal Process. Lett.* **2015**, *22*, 2450–2454.
13. Sarkka, S.; Nummenmaa, A. Recursive Noise Adaptive Kalman Filtering by Variational Bayesian Approximations. *IEEE Trans. Autom. Control* **2009**, *54*, 596–600.
14. Qi, S.; Han, J.D. An Adaptive UKF Algorithm for the State and Parameter Estimations of a Mobile Robot. *Acta Autom. Sinica* **2008**, *34*, 72–79.
15. Wang, J. Stochastic Modeling for Real-Time Kinematic GPS/GLONASS Positioning. *Navigation* **1999**, *46*, 297–305.
16. Assa, A.; Plataniotis, K.N. Adaptive Kalman Filtering by Covariance Sampling. *IEEE Signal Process. Lett.* **2017**, *24*, 1288–1292.
17. Zanni, L.; Boudec, J.Y.L.; Cherkaoui, R.; Paolone, M. A Prediction-Error Covariance Estimator for Adaptive Kalman Filtering in Step-Varying Processes: Application to Power-System State Estimation. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 1683–1697.
18. Ghaleb, F.A.; Zainal, A.; Rassam, M.A.; Abraham, A. Improved vehicle positioning algorithm using enhanced innovation-based adaptive Kalman filter. *Perv. Mob. Comput.* **2017**, *40*, 139–155.
19. Hajiyev, C.; Soken, H.E. Robust adaptive unscented Kalman filter for attitude estimation of pico satellites. *Int. J. Adapt. Control Signal Process.* **2014**, *28*, 107–120.
20. Zhe, J.; Qi, S.; Yuqing, H.; Jianda, H. A novel adaptive unscented Kalman filter for nonlinear estimation. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 4293–4298.
21. Shi, Y.; Han, C.; Liang, Y. Adaptive UKF for target tracking with unknown process noise statistics. In Proceedings of the International Conference on Information Fusion, Seattle, WA, USA, 6–9 July 2009; pp. 1815–1820.
22. Gao, S.; Hu, G.; Zhong, Y. Windowing and random weighting-based adaptive unscented Kalman filter. *Int. J. Adapt. Control Signal Process.* **2015**, *29*, 201–223.
23. Zhou, J.; Knedlik, S.; Loffeld, O. INS/GPS tightly-coupled integration using adaptive unscented particle filter. *J. Navig.* **2010**, *63*, 491–511.

24. Soken, H.E.; Hajiyev, C. A Novel Adaptive Unscented Kalman Filter for Pico Satellite Attitude Estimation. In Proceedings of the 5th International Conference on Physics and Control, León, Spain, 5–8 September 2011.
25. Soken, H.E.; Hajiyev, C. UKF-Based Reconfigurable Attitude Parameters Estimation and Magnetometer Calibration. *IEEE Trans. Aerospace Electr. Syst.* **2012**, *48*, 2614–2627.
26. Li, W.; Sun, S.; Jia, Y.; Du, J. Robust unscented Kalman filter with adaptation of process and measurement noise covariances. *Digit. Signal Process.* **2016**, *48*, 93–103.
27. Das, M.; Sadhu, S.; Ghoshal, T.K. Spacecraft attitude & rate estimation by an adaptive unscented Kalman filter. In Proceedings of the 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC), Calcutta, India, 31 January–2 February 2014; pp. 46–50.
28. Wan, E.A.; Merwe, R.V.D. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
29. Davari, N.; Gholami, A. An Asynchronous Adaptive Direct Kalman Filter Algorithm to Improve Underwater Navigation System Performance. *IEEE Sens. J.* **2017**, *17*, 1061–1068.
30. Mohamed, A.H.; Schwarz, K.P. Adaptive Kalman Filtering for INS/GPS. *J. Geod.* **1999**, *73*, 193–203.
31. Zhan, R.; Wan, J. Iterated Unscented Kalman Filter for Passive Target Tracking. *IEEE Trans. Aerospace Electr. Syst.* **2007**, *43*, 1155–1163.
32. Rouhani, A.; Abur, A. Constrained Iterated Unscented Kalman Filter for Dynamic State and Parameter Estimation. *IEEE Trans. Power Syst.* **2017**, *PP*, 1-1.
33. Li, X.R.; Zhao, Z. Evaluation of estimation algorithms part I: incomprehensive measures of performance. *IEEE Trans. Aerospace Electr. Syst.* **2006**, *42*, 1340–1358.
34. Fu, P.; Cheng, Y.; Tang, H.; Li, B.; Pei, J.; Yuan, X. An Effective and Robust Decentralized Target Tracking Scheme in Wireless Camera Sensor Networks. *Sensors* **2017**, *17*, 639.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).