applied sciences

MDPI

Article

Robust Estimation in Continuous–Discrete Cubature Kalman Filters for Bearings-Only Tracking

Haoran Hu [1], Shuxin Chen [1], Hao Wu [1,2,*] and Renke He [3]

[1] Information and Navigation College, Air Force Engineering University, Xi'an 710077, China
[2] State Key Laboratory of Geo-Information Engineering, Xi'an Research Institute of Survey and Mapping, Xi'an 710054, China
[3] Science and Technology Complex Aviation Systems Simulation Laboratory, Beijing 100076, China
* Correspondence: wuhaostudy@163.com; Tel.: +86-029-8479-1616

Abstract: The model of bearings-only tracking is generally described by discrete–discrete filtering systems. Discrete robust methods are also frequently used to address measurement uncertainty problems in bearings-only tracking. The recently popular continuous–discrete filtering system considers the state model of the target to be continuous in time, and is more suitable for bearings-only tracking because of its higher mathematical solution accuracy. However, the sufficient evaluation of robust methods in continuous–discrete systems is not available. In addition, in the different continuous–discrete measurement environments, the choice of a robust algorithm also needs to be discussed. To fill this gap, this paper firstly establishes the continuous–discrete target tracking model, and then evaluates the performance of proposed robust square-root continuous–discrete cubature Kalman filter algorithms in the measurement of uncertainty problems. From the simulation results, the robust square-root continuous–discrete maximum correntropy cubature Kalman filter algorithm and the variational Bayesian square-root continuous–discrete cubature Kalman filter algorithm have better environmental adaptability, which provides a promising means for solving continuous–discrete robust problems.

Keywords: cubature Kalman filter; bearings-only tracking; continuous–discrete time system; non-gaussian noise; maximum correntropy; variational Bayesian

check for updates

Citation: Hu, H.; Chen, S.; Wu, H.; He, R. Robust Estimation in Continuous–Discrete Cubature Kalman Filters for Bearings-Only Tracking. *Appl. Sci.* **2022**, *12*, 8167. https://doi.org/10.3390/app12168167

Academic Editor: Giovanni Petrone

Received: 12 July 2022
Accepted: 13 August 2022
Published: 15 August 2022

1. Introduction

Bearings-only tracking [1,2] uses a sequence of angle measurements to estimate the state (position and velocity) of the target and has universal applications in the field of navigation, especially in passive target tracking. For the system model of bearings-only tracking, the target state is generally built as the discrete time model for convenience. In addition, the angle measurement information for the target is discrete time. This is called the discrete–discrete time system model. However, considering the actual situation of target tracking, the target state is continuous in nature, and the measurement is discrete time. Such a system model is called the continuous–discrete time system [1]. Compared with the discrete–discrete system model, the continuous–discrete system model is closer to the real situation, and its solutions have more advantages, such as high accuracy [3].

Based on the continuous–discrete time system, continuous–discrete filtering, whose model is based on the covariance matrix of random errors, can be established. In [4], the cubature criterion was introduced into the continuous–discrete time system, and the continuous–discrete cubature Kalman filter (CD-CKF) algorithm was obtained. It uses stochastic differential equations (SDEs) to describe the continuous-time target state model, and the accuracy of the SDE solution greatly affects the filtering performance. In [5], Crouse summarized the Euler-Maruyama method based on 0.5 order and the Itô-Taylor method, which were proven to have acceptable accuracy. In order to further improve the accuracy of state estimation, high-order numerical approximation methods are used to deal with stochastic differential equations. In [6], Crouse described the continuous-time model as the form of its expectation and covariance and then solved the time prediction problem of the continuous model. In [7], the nested implicit Runge-Kutta method was used for the continuous–discrete extended Kalman filter (CD-EKF), and the corresponding application in CD-CKF was discussed in [8]. Subsequently, adaptive methods were proposed. In [9], Kulikova et al. used adaptive step size to further improve the accuracy. The adaptive feedback strategy [10] reduced the impact of unpredictable errors in the prediction step by means of covariance adjustment; thus, the performance and efficiency were both improved.

In the actual measurement environment, harsh conditions such as signal interference will cause non-Gaussian noise [11,12] or outliers to appear. If they are not dealt with, the Gaussian-assumed filters will be inaccurate or even divergent. For this measurement uncertainty problem, various robust methods [13] are proposed for the discrete–discrete time filtering systems. As a method of modification, robust methods aim to improve robustness in the presence of outliers in the data. For instance, the Huber's *M*-estimation method [14,15] used the robust cost function to reduce the weight of abnormal data. It is widely used and has proven effective in filtering problems [16]. Wu et al. [17,18] proposed a more practical robust CKF algorithm based on generalized *M*-estimation without the experience threshold of the weight function and statistical characteristics information of outliers. In [19], Särkkä dealt with the time-varying noise with the variational Bayes criterion in the case of unknown noise covariance. In [20–22], the maximum correntropy criterion (MCC) was introduced into discrete–discrete filtering systems. The correntropy can utilize the higher-order information of the measurements instead of the second-order moment information of the minimum mean square error criterion, which is more suitable for non-Gaussian systems in the presences of non-Gaussian noise.

The above robust methods are successfully used in the discrete–discrete time systems, but sufficient evaluation of the continuous–discrete time systems is not available. Moreover, how to balance the accuracy and robust performance of the continuous–discrete filtering algorithms is of great significance. Therefore, this paper proposes the square-root continuous–discrete cubature Kalman filter (SRCD-CKF) with Huber's method, maximum correntropy criterion, and variational Bayesian criterion, and analyzes their filtering performance in the measurement of uncertainty problems. The CD-CKF is chosen because it is derivative-free and generally more accurate than the CD-EKF. Additionally, compared with the continuous–discrete unscented Kalman filter (CD-UKF), CD-CKF has the smaller computational burden. The square-root technique can also improve the stability of the filter. Compared with the proposed robust algorithms, the robust square-root continuous–discrete maximum correntropy cubature Kalman filter (RSRCD-MCCKF) algorithm and the variational Bayesian square-root continuous–discrete cubature Kalman filter (VBSRCD-CKF) algorithm have higher accuracy and stronger robustness.

To sum up, in this paper, the robust estimation methods based on a continuous–discrete system are proposed to solve the problem of sufficient evaluation of continuous–discrete robust algorithms. Compared with the discrete–discrete filtering system, considering the continuous–discrete filtering system can effectively ensure the accuracy of bearings-only tracking, while the robust methods are used to improve the robustness of the tracking. From the perspective of improving accuracy and robustness, the proposed algorithms have excellent performance in continuous–discrete systems. Another contribution is that the article compares the performance of the algorithms in different continuous–discrete measurement environments, fully considering the environmental adaptability of the algorithms.

The structure of the rest of the article is as follows: Section 2 reviews the target tracking model of the continuous–discrete system and the algorithm of SRCD-CKF, Section 3 proposes the robust SRCD-CKF algorithms with Huber's method, the RSRCD-MCCKF algorithm, and the VBSRCD-CKF algorithm, Section 4 shows the results of the simulation, and Section 5 summarizes the main work of this paper.

2. Continuous–Discrete System Model and Square-Root Continuous–Discrete Cubature Kalman Filter Algorithm for Target Tracking

2.1. Continuous–Discrete System Model

In a continuous–discrete target tracking system, the target state model is continuous time, and the measurement model is discrete time. Considering the random disturbance, the continuous-time target state model can be described as the following stochastic differential equation form [1]:

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t)dt + \sqrt{\boldsymbol{Q}}d\boldsymbol{w}(t), \tag{1}$$

where $\boldsymbol{x}(t)$ is the n-dimensional state vector, $\boldsymbol{f}$ is known as the drift function, $\boldsymbol{w}(t)$ is the zero-mean Gaussian white-noise process, and $\boldsymbol{Q}$ represents the covariance matrix of the zero-mean Gaussian white-noise process.

In order to deal with the continuous time model, the target model of the continuous–discrete system can be described by the moment differential equations (MDEs) [9], which are shown as

$$\frac{d\hat{\boldsymbol{x}}(t)}{dt} = \boldsymbol{F}(\hat{\boldsymbol{x}}(t)), \tag{2}$$

$$\frac{d\boldsymbol{P}(t)}{dt} = \boldsymbol{J}(\hat{\boldsymbol{x}}(t))\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{J}^{\mathrm{T}}(\hat{\boldsymbol{x}}(t)) + \boldsymbol{Q}, \tag{3}$$

where $\hat{\boldsymbol{x}}(t)$ is the expectation of $\boldsymbol{x}(t)$ at time t, $\boldsymbol{P}(t)$ represents the error covariance matrix at time t, $\boldsymbol{F}(\hat{\boldsymbol{x}}(t))$ is the state equation of the system, and the Jacobian matrix of $\boldsymbol{F}(\hat{\boldsymbol{x}}(t))$ is $\boldsymbol{J}(\hat{\boldsymbol{x}}(t))$, that is, $\boldsymbol{J}(\hat{\boldsymbol{x}}(t)) = \partial\boldsymbol{F}(\hat{\boldsymbol{x}}(t))/\partial\hat{\boldsymbol{x}}(t)$.

The discrete time measurement model is

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, k) + \boldsymbol{r}_k, \tag{4}$$

where $\boldsymbol{z}_k$ is the actual observation value, $\boldsymbol{h}$ is the observation function, and $\boldsymbol{r}_k$ is the observation noise, which refers to Gaussian or non-Gaussian noise. The specific noise form of $\boldsymbol{r}_k$ is defined by the measurement scenario, and its covariance matrix is $\boldsymbol{R}_k$. Here, k refers to discrete time measurement points.

2.2. Square-Root Continuous–Discrete Cubature Kalman Filter Algorithm

In order to ensure the positive definiteness and symmetry of the error covariance matrix, the square-root continuous–discrete cubature Kalman filter is introduced here. Similar to the typical CKF method, the continuous–discrete time system is combined with the cubature criterion, which is also divided into two steps: time update and measurement update.

In the SRCD-CKF algorithm, the state cubature points can be defined as [23]

$$\boldsymbol{X}_i(t) = \boldsymbol{S}(t)\xi_i + \hat{\boldsymbol{x}}(t), \tag{5}$$

where $\boldsymbol{S}(t)$ is the lower triangular matrix of the covariance matrix, satisfying $\boldsymbol{P}(t) = \boldsymbol{S}(t)\boldsymbol{S}^{\mathrm{T}}(t)$, and ξ_i is the cubature point set, which is defined as

$$\xi_i(t) = \begin{cases} l\sqrt{n}\boldsymbol{\eta}_i & i = 1, \cdots, n \\ \sqrt{n}\boldsymbol{\eta}_{i-n} & i = n+1, \cdots, 2n \end{cases} \tag{6}$$

where $\boldsymbol{\eta}_i$ is the i-th coordinate column-vector in $\mathbb{R}^n$.

Based on the cubature criterion, its expectation and covariance can be re-expressed as

$$\frac{d\hat{\boldsymbol{x}}(t)}{dt} = \boldsymbol{F}(\boldsymbol{X}(t))\boldsymbol{\varepsilon}, \tag{7}$$

$$\frac{d\boldsymbol{P}(t)}{dt} = \boldsymbol{X}(t)\boldsymbol{W}\boldsymbol{F}^{\mathrm{T}}(\boldsymbol{X}(t)) + \boldsymbol{F}(\boldsymbol{X}(t))\boldsymbol{X}^{\mathrm{T}}(t) + \boldsymbol{Q}(t). \tag{8}$$

The parameters of $\boldsymbol{\varepsilon}$ and $\boldsymbol{W}$ are defined as follows:

$$\begin{cases} \boldsymbol{\varepsilon} = \frac{\mathbf{1}}{2n} \\ \boldsymbol{W} = (\boldsymbol{I}_{2n} - \mathbf{1}^{\mathrm{T}} \otimes \boldsymbol{\varepsilon})\mathrm{diag}[1/(2n), \cdots, 1/(2n)](\boldsymbol{I}_{2n} - \mathbf{1}^{\mathrm{T}} \otimes \boldsymbol{\varepsilon})^{\mathrm{T}}, \end{cases} \tag{9}$$

where $\boldsymbol{I}_{2n}$ represents a unit matrix with a dimension of $2n$, $\mathbf{1}$ is a unit column vector, and $\otimes$ represents the Kronecker tensor.

For the calculation of covariance propagation, the square-root covariance calculation method proposed in [24] is used. Meanwhile, the high-order numerical approximation method is used to deal with the problem:

$$\frac{d\boldsymbol{S}(t)}{dt} = \boldsymbol{S}(t)\boldsymbol{\Phi}(\boldsymbol{B}), \tag{10}$$

$$\boldsymbol{\Phi}(\boldsymbol{B}) = \begin{cases} \boldsymbol{B}_{i,j} & \mathit{if}\ i > j \\ \frac{1}{2}\boldsymbol{B}_{i,j} & \mathit{if}\ i = j, \\ 0 & \mathit{if}\ i < j \end{cases} \tag{11}$$

where $\boldsymbol{B}_{i,j}$ is the element in the i-th row and j-th column of $\boldsymbol{\Phi}(\boldsymbol{B})$. The definition of $\boldsymbol{B}(t)$ matrix is

$$\boldsymbol{B}(t) = \boldsymbol{S}^{-1}(t)\Big[\boldsymbol{X}(t)\boldsymbol{W}\boldsymbol{F}^{\mathrm{T}}(\boldsymbol{X}(t)) + \boldsymbol{F}(\boldsymbol{X}(t))\boldsymbol{X}^{\mathrm{T}}(t) + \boldsymbol{Q}(t)\Big]\boldsymbol{S}^{-\mathrm{T}}(t). \tag{12}$$

Based on this, the predicted state cubature point $\boldsymbol{X}_i^*(t)$ can be obtained. The specific SRCD-CKF Algorithm 1 is as follows:

```
Algorithm 1: SRCD-CKF.
Time update
Step 1 Expectation and covariance matrix initialization: $\hat{\boldsymbol{x}}(t_k) = \hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{P}(t_k) = \boldsymbol{P}_{k-1|k-1}$.
Step 2.1 Covariance decomposition: $\boldsymbol{P}(t_k) = \boldsymbol{S}(t_k)\boldsymbol{S}^{\mathrm{T}}(t_k)$.
Step 2.2 Calculate the state cubature point: $\boldsymbol{X}_i(t_k) = \boldsymbol{S}(t_k)\xi_i + \hat{\boldsymbol{x}}(t_k)$.
Step 2.3 State cubature point propagation: $\begin{cases} \hat{\boldsymbol{x}}\prime(t) = \boldsymbol{F}(\boldsymbol{X}(t))\boldsymbol{\varepsilon} \\ \boldsymbol{S}\prime(t) = \boldsymbol{S}(t)\boldsymbol{\Phi}(\boldsymbol{B}(t)) \end{cases}$.
Step 2.4 Calculate the state prediction value: $\hat{\boldsymbol{x}}_{k|k-1} = \left(\boldsymbol{X}_1^*(t_k) + \cdots + \boldsymbol{X}_{2n}^*(t_k)\right)/2n$.
Step 2.5 Calculate the predicted square-root covariance:
    $\boldsymbol{S}(t_k) = \begin{bmatrix}\boldsymbol{S}_1(t_k) & \boldsymbol{S}_2(t_k) & \cdots & \boldsymbol{S}_{2n}(t_k)\end{bmatrix}$,
where $\boldsymbol{S}_i(t_k)$ represents $a$ sin gle column vector, $\boldsymbol{S}_i(t_k) = \left(\boldsymbol{X}_i^*(t_k) - \hat{\boldsymbol{x}}_{k|k-1}\right)/\sqrt{\mathrm{n}}$.
Measurement update
Step 3.1 Calculate the state cubature point: $\boldsymbol{S}_{k|k-1} = \boldsymbol{S}(t_k), \boldsymbol{X}_{i,k|k-1} = \boldsymbol{S}_{k|k-1}\xi_i + \hat{\boldsymbol{x}}_{k|k-1}$.
Step 3.2 Measure the spread of cubature points: $\boldsymbol{Z}_{i,k|k-1} = h(\boldsymbol{X}_{i,k|k-1}, k)$.
Step 3.3 Calculate the predicted value of the measurement: $\hat{\boldsymbol{z}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\boldsymbol{Z}_{i,k|k-1}$.
Step 3.4 Construct the measurement weighted center matrix:
    $\boldsymbol{Z}_{k|k-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{Z}_{1,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}, \cdots, \boldsymbol{Z}_{2n,k|k-1} - \hat{\boldsymbol{z}}_{k|k-1}\right]$.
Step 3.5 Calculate the innovation covariance matrix: $\boldsymbol{P}_{zz,k|k-1} = \boldsymbol{Z}_{k|k-1}\boldsymbol{Z}_{k|k-1}^{T} + \boldsymbol{R}_k$.
Step 3.6 Construct the state weighted center matrix:
    $\boldsymbol{X}_{k|k-1} = \frac{1}{\sqrt{2n}}\left[\boldsymbol{X}_{1,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}, \cdots, \boldsymbol{X}_{2n,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}\right]$.
Step 3.7 Calculate the cross covariance matrix: $\boldsymbol{P}_{xz,k|k-1} = \boldsymbol{X}_{k|k-1}\boldsymbol{Z}_{k|k-1}^{\mathrm{T}}$.
Step 3.8 The continuous–discrete cubature gain is: $\boldsymbol{K}_k = \boldsymbol{P}_{xz,k|k-1}\boldsymbol{P}_{zz,k|k-1}^{-1}$.
Step 3.9 Calculate the state estimate: $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})$.
Step 3.10 Update the covariance matrix: $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k P_{zz,k|k-1}\boldsymbol{K}_k^{\mathrm{T}}$.
```

Remark 1. *The above method gives the square root form of the prediction covariance and uses it for the measurement update process. The algorithm has effectively guaranteed the stability of the filtering. At the same time, a more thorough square-root version is also available.*

Here we redefine the cross-covariance matrix $\boldsymbol{P}_{xz,k|k-1}$, the innovations covariance square-root $\boldsymbol{R}_{e,k}^{1/2}$, and the filtering covariance square-root $\boldsymbol{S}_{k|k}$ by QR decomposition:

$$\mathrm{qr}\begin{bmatrix} \boldsymbol{Z}_{k|k-1} & \boldsymbol{R}_k^{1/2} \\ \boldsymbol{X}_{k|k-1} & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{e,k}^{1/2} & 0 \\ \boldsymbol{P}_{xz,k|k-1} & \boldsymbol{S}_{k|k} \end{bmatrix}, \tag{13}$$

where **0** is the zero-block of a proper size, $\boldsymbol{R}_k^{1/2}$ is the measurement noise covariance square-root obtained by Cholesky factorization, and $\boldsymbol{R}_{e,k}^{1/2}$ and $\boldsymbol{S}_{k|k}$ are lower triangular matrices.

On this basis, we can obtain the update state and covariance. In addition, the robust estimation methods under this square-root version can be obtained according to our idea.

3. Robust Square-Root Continuous–Discrete Cubature Kalman Filter Algorithms

3.1. Robust SRCD-CKF with Huber's Method

The robust SRCD-CKF algorithms are used to correct the abnormal measurements. Huber's method uses the robust cost function to reduce the weight of abnormal data. Among them, the most common weight reduction functions are the two-stage weight function based on Huber and the three-stage weight function based on *p*-Huber. The Huber weight function usually includes two parts. This means that the measurement data are divided into normal data and abnormal data. When the absolute value of the standardized residual exceeds the critical value, the data are considered abnormal and the weight is reduced.

$$\overline{\mu}_k = \begin{cases} 1, \left|\widetilde{\lambda}_{\mathrm{b}}\right| < \mathrm{c} \\ \\ \mathrm{c}/\left|\widetilde{\lambda}_{\mathrm{b}}\right|, \left|\widetilde{\lambda}_{\mathrm{b}}\right| \geq \mathrm{c} \end{cases}, \tag{14}$$

where $\left|\widetilde{\lambda}_{\mathrm{b}}\right|$ is the absolute value of the standardized residuals, and c is the critical value of abnormal error discrimination. The Huber robust square-root continuous–discrete cubature Kalman filter (HRSRCD-CKF) Algorithm 2 is as follows:

```
Algorithm 2: HRSRCD-CKF.
Step 1 Repeat Step 1 of the SRCD-CKF algorithm.
Steps 2.1–3.7 are equivalent to Steps 2.1–3.7 of the SRCD-CKF algorithm.
Step 3.8 The innovation covariance matrix is redefined: $\overline{\boldsymbol{P}}_{zz,k|k-1} = \boldsymbol{Z}_{k|k-1}\boldsymbol{Z}_{k|k-1}^{\mathrm{T}} + \overline{\mu}_k^{-1}\boldsymbol{R}_k$.
Step 3.9–3.11 are equivalent to Steps 3.8–3.10 of the SRCD-CKF algorithm.
```

The weight function based on the Mahalanobis distance is also called the *p*-Huber three-stage weight function. Compared with the Huber function, the *p*-Huber function uses the α quantile of the chi-square distribution to adjust the critical value of abnormal error discrimination, which is more reasonable than the empirical value. The discriminant value is defined as follows:

$$d_k = \mathrm{M}_k^2 = (\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}}\left(\boldsymbol{P}_{zz,k|k-1}\right)^{-1}(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}). \tag{15}$$

In Equation (15), $\mathrm{M}_k = \sqrt{(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})^{\mathrm{T}}\left(\boldsymbol{P}_{zz,k|k-1}\right)^{-1}(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1})}$ is the Mahalanobis distance. The new three-stage weight function is

$$\overline{\mu}_k = \begin{cases} 1, d_k < \chi_{\alpha 1} \\ \\ \sqrt{\chi_{\alpha 1}/d_k}, \chi_{\alpha 1} \leq d_k < \chi_{\alpha 2}, \\ \\ 0, d_k \geq \chi_{\alpha 2} \end{cases} \tag{16}$$

where χ_α is the quantile of the chi-square distribution. When d_k exceeds $\chi_{\alpha 1}$, it is considered that there is an outlier in the measurement, and its weight needs to be reduced. At the same time, in order to further eliminate the influence of large outliers, when $d_k \geq \chi_{\alpha 2}$, they will be eliminated. The M-estimation–based robust square-root continuous–discrete cubature Kalman filter (MRSRCD-CKF) Algorithm 3 is as follows:

Algorithm 3: MRSRCD-CKF.

Step 1 Repeat Step 1 of the SRCD-CKF algorithm.
Steps 2.1–3.7 are equivalent to Steps 2.1–3.7 of the SRCD-CKF algorithm.
Steps 3.8–3.11 are equivalent to Steps 3.8–3.11 of the HRSRCD-CKF algorithm, and $\overline{\mu}_k$ is calculated by Equation (16).

3.2. RSRCD-MCCKF Algorithm Based on Maximum Correntropy Criterion

Correntropy is a new method used to measure the similarity between two random variables. Given the random variables X and Y, their joint probability density function is $F_{X,Y}(x,y)$. The correntropy can be defined as follows [22]:

$$V(X,Y) = E[\kappa(X,Y)] = \iint \kappa(x,y)\mathrm{d}F_{X,Y}(x,y), \tag{17}$$

where $E[\cdot]$ is the mathematical expectation, and $\kappa(\cdot)$ is the Messer-type positive definite kernel function. In this paper, we select the Gaussian kernel as the kernel function of the correntropy, which is defined as follows:

$$\kappa(x,y) = G_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right), \tag{18}$$

where $e = x - y$, and $\sigma > 0$ is the kernel bandwidth of the kernel function.

In practice, the joint probability density function is usually difficult to obtain, and the data sampling points are also limited. Therefore, the correntropy can be estimated by the approach of T data sampling points:

$$\hat{V}(X,Y) = \frac{1}{T}\sum_{i=1}^{T} G_\sigma(e(i)). \tag{19}$$

where $e(i) = x(i) - y(i)$, $\{x(i), y(i)\}_{i=1}^{T}$ represents the related sampling point sequence of the joint probability density function $F_{X,Y}(x,y)$. Expanding the Gaussian kernel function using the Taylor series, we can obtain

$$V(X,Y) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n \sigma^{2n} n!} E\left[(X-Y)^{2n}\right]. \tag{20}$$

It can be seen from Equation (20) that the correntropy information contains all the even-order moments of the random variable error $X{-}Y$. Therefore, as long as the appropriate kernel bandwidth is selected, higher-order information can be obtained. For the systems contaminated by noise, compared with the minimum mean square error criterion, the correntropy can be used to describe its statistical information.

On this basis, this paper introduces MCC into the continuous–discrete system. Firstly, $\overline{\boldsymbol{H}}_k$ is defined as the pseudo-measurement matrix, $\overline{\boldsymbol{R}}_k$ is the estimated covariance matrix of the observation noise, and their calculation methods are as follows:

$$\overline{\boldsymbol{H}}_k = \boldsymbol{P}_{xz,k|k-1}^{T} \boldsymbol{P}_{k|k-1}^{-1}, \tag{21}$$

$$\overline{\boldsymbol{R}}_k = \boldsymbol{P}_{zz,k|k-1} - \overline{\boldsymbol{H}}_k \boldsymbol{P}_{k|k-1} \overline{\boldsymbol{H}}_k^{\mathrm{T}}. \tag{22}$$

The cost function based on the MCC filtering can be defined by

$$J_{\mathrm{MCC}} = G_\sigma\left(\left\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}\right\|^2_{\boldsymbol{P}^{-1}_{k|k-1}}\right) + G_\sigma\left(\left\|\boldsymbol{z}_k - \overline{\boldsymbol{H}}_k\boldsymbol{x}_k - \hat{\boldsymbol{z}}_{k|k-1} + \overline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k|k-1}\right\|_{\overline{\boldsymbol{R}}^{-1}_k}\right). \tag{23}$$

In this paper, the process noise is assumed to be Gaussian, and we modify the cost function as follows:

$$J_{\mathrm{MCC}} = \gamma\left\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}\right\|^2_{\boldsymbol{P}^{-1}_{k|k-1}} - \beta G_\sigma\left(\left\|\boldsymbol{z}_k - \overline{\boldsymbol{H}}_k\boldsymbol{x}_k - \hat{\boldsymbol{z}}_{k|k-1} + \overline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k|k-1}\right\|_{\overline{\boldsymbol{R}}^{-1}_k}\right), \tag{24}$$

where γ and β are adjusting weights, and $\|\boldsymbol{x}\|^2_{\boldsymbol{A}} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$. Then, the optimal estimation of $\boldsymbol{x}_k$ is

$$\hat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}_k}{\mathrm{argmin}} J_{\mathrm{MCC}}. \tag{25}$$

The optimal solution of the above equation can be obtained by solving the derivative

$$\frac{\partial J_{\mathrm{MCC}}}{\partial \boldsymbol{x}_k} = \gamma\boldsymbol{P}^{-1}_{k|k-1}\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}\right) - \beta\frac{\mathrm{L}_k\overline{\boldsymbol{H}}^{\mathrm{T}}_k\overline{\boldsymbol{R}}^{-1}_k}{-2\sigma^2}\left(\boldsymbol{z}_k - \overline{\boldsymbol{H}}_k\boldsymbol{x}_k - \hat{\boldsymbol{z}}_{k|k-1} + \overline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k|k-1}\right) = 0 \tag{26}$$

With

$$\mathrm{L}_k = G_\sigma\left(\left\|\boldsymbol{z}_k - \overline{\boldsymbol{H}}_k\boldsymbol{x}_k - \hat{\boldsymbol{z}}_{k|k-1} + \overline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k|k-1}\right\|_{\overline{\boldsymbol{R}}^{-1}_k}\right). \tag{27}$$

This correntropy criterion follows the idea of *M*-estimation, applying the Gaussian kernel into each element of the estimation error matrix. The adjustment factor obtained by the Gaussian kernel function (denoted as L_k) will be a scalar, which is easily separated and avoids the numerical calculation problem of zero matrix inversion.

When $\sigma \to \infty$, in order to guarantee the algorithm can converge to the traditional CD-CKF algorithm, we set $\gamma = 1$, $\beta = -2\sigma^2$ in this paper. Then, the state estimate and the new filter gain can be obtained by the fixed-point iteration method:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \overline{\overline{\boldsymbol{K}}}_k\left(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}\right), \tag{28}$$

$$\overline{\overline{\boldsymbol{K}}}_k = \mathrm{L}_k\boldsymbol{P}_{k|k-1}\overline{\boldsymbol{H}}^{\mathrm{T}}_k\left(\overline{\boldsymbol{R}}_k + \mathrm{L}_k\overline{\boldsymbol{H}}_k\boldsymbol{P}_{k|k-1}\overline{\boldsymbol{H}}^{\mathrm{T}}_k\right)^{-1}. \tag{29}$$

The update of the error covariance matrix can be calculated by the following equation:

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \overline{\overline{\boldsymbol{K}}}_k\overline{\boldsymbol{H}}_k)\boldsymbol{P}_{k|k-1}. \tag{30}$$

Considering that in the fixed-point iteration, $\overline{\boldsymbol{H}}_k\boldsymbol{x}_k$ can be replaced by $\overline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k|k-1}$ [20], L_k can be rewritten as

$$L_k = G_\sigma\left(\left\|\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}\right\|_{\overline{\boldsymbol{R}}^{-1}_k}\right), \tag{31}$$

where $G_\sigma(\cdot)$ can be calculated by Equation (18).

Therefore, the RSRCD-MCCKF Algorithm 4 can be summarized as follows:

Algorithm 4: RSRCD-MCCKF.

Step 1 Repeat Step 1 of the SRCD-CKF algorithm.
Steps 2.1–3.7 are equivalent to Steps 2.1–3.7 of the SRCD-CKF algorithm.
Step 3.8 Calculate $\overline{\boldsymbol{H}}_k$, $\overline{\boldsymbol{R}}_k$, L_k using Equations (21), (22), and (31).
Step 3.9 Obtain the new Kalman filter gain $\overline{\overline{\boldsymbol{K}}}_k$ using Equation (29).
Step 3.10 Complete the estimation of the state value and the update of the error covariance matrix using Equations (28) and (30).

In the RSRCD-MCCKF, note that the Gaussian kernel bandwidth σ greatly affects the accuracy of the filtering algorithm. Studies have shown that [25] when σ is small, it is more robust to non-Gaussian measurement noise; however, when σ is too small, the filtering performance of the algorithm will be more severely degraded, and the filtering may even diverge. For the actual situation of continuous–discrete systems, how to select the proper value of σ according to different scenarios is also of great significance.

3.3. VBSRCD-CKF Algorithm Based on Variational Bayes Criterion

Variational Bayesian approximation is an iterative optimization algorithm that approximates the posterior probability distribution of parameters [26]. The VB-based filter uses VB approximation to estimate the joint posterior distribution of the state and covariance $p(\boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})$, as follows:

$$p(\boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k}) \approx \boldsymbol{Q}_x(\boldsymbol{x}_k)\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k), \tag{32}$$

where $\boldsymbol{Q}_x(\boldsymbol{x}_k)$ and $\boldsymbol{Q}_R(\boldsymbol{R}_k)$ are the yet unknown approximating densities.

Unlike the correntropy, the Kullback-Leibler (KL) divergence compares the closeness of two probability distributions. It is now possible to form a VB approximation by minimizing the KL divergence between the separable approximation and the true posterior:

$$\mathrm{KL}[\boldsymbol{Q}_x(\boldsymbol{x}_k)\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k)\|\mathrm{p}(\boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})] = \int \boldsymbol{Q}_x(\boldsymbol{x}_k)\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k) \times \log\left(\frac{\boldsymbol{Q}_x(\boldsymbol{x}_k)\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k)}{p(\boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})}\right)d\boldsymbol{x}_k d\boldsymbol{R}_k. \tag{33}$$

Minimizing the KL divergence associated with the probability densities $\boldsymbol{Q}_x(\boldsymbol{x}_k)$ and $\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k)$ in turn, while keeping the other fixed, we can obtain the following equations:

$$\boldsymbol{Q}_x(\boldsymbol{x}_k) \propto \exp\left(\int \log p(\boldsymbol{z}_k, \boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k)d\boldsymbol{R}_k\right), \tag{34}$$

$$\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k) \propto \exp\left(\int \log p(\boldsymbol{z}_k, \boldsymbol{x}_k, \boldsymbol{R}_k|\boldsymbol{z}_{1:k})\boldsymbol{Q}_x(\boldsymbol{x}_k)d\boldsymbol{x}_k\right). \tag{35}$$

Computing the above equations, we can obtain the following densities [19]:

$$\boldsymbol{Q}_x(\boldsymbol{x}_k) = N(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k), \tag{36}$$

$$\boldsymbol{Q}_\mathrm{R}(\boldsymbol{R}_k) = \mathrm{IW}(\boldsymbol{R}_k|v_k, \boldsymbol{V}_k), \tag{37}$$

where IW(·) represents the inverse Wishart (IW) distribution, and the parameters $\hat{\boldsymbol{x}}_k$, $\boldsymbol{P}_k$, v_k and $\boldsymbol{V}_k$ can be calculated by Kalman-type filters. The VBSRCD-CKF Algorithm 5 is as follows:

Algorithm 5: VBSRCD-CKF.

```
Step 1 At Step 1 of the SRCD-CKF algorithm, the initialization of $v_k$ and $\boldsymbol{V}_k$ are included.
Steps 2.1–2.5 are equivalent to Steps 2.1–2.5 of the SRCD-CKF algorithm.
Step 2.6 Calculate the parameters of the IW distribution of measurement noise covariance: $v_{k|k-1} = \rho(v_{k-1} - n - 1) + n + 1$, $\boldsymbol{V}_{k|k-1} = \boldsymbol{C}\boldsymbol{V}_{k-1}\boldsymbol{C}^T$, where ρ is a scale factor that 0 < ρ ≤ 1 and $\boldsymbol{C}$ is a matrix that $0 < |\boldsymbol{C}| \leq 1$ with a reasonable choice for the matrix $\boldsymbol{C} = \sqrt{\rho}\boldsymbol{I}_d$. $\boldsymbol{I}_d$ is an identity matrix, and $d$ is the dimension of the measurement.
Step 3.1 Before calculating the state cubature point, first let $\hat{\boldsymbol{x}}_k^{(0)} = \hat{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{V}_k^{(0)} = \boldsymbol{V}_{k|k-1}$, and $v_k = 1 + v_{k|k-1}$.
Steps 3.2–3.8 are equivalent to Steps 3.1–3.7 of the SRCD-CKF algorithm.
Step 3.9 For j = 1:M, iterate the following steps (M is the times of algorithm iteration).
Step 3.9.1 Calculate the measurement noise covariance matrix: $\boldsymbol{R}_k^{(j)} = (v_k - n - 1)^{-1}\boldsymbol{V}_k^{(j-1)}$.
Step 3.9.2 Update the innovation covariance matrix: $\boldsymbol{P}_{zz,k|k-1}^{(j)} = \boldsymbol{Z}_{k|k-1}\boldsymbol{Z}_{k|k-1}^\mathrm{T} + \boldsymbol{R}_k^{(j)}$.
Step 3.9.3 Calculate the continuous–discrete filter gain: $\boldsymbol{K}_k^{(j)} = \boldsymbol{P}_{xz,k|k-1}\left(\boldsymbol{P}_{zz,k|k-1}^{(j)}\right)^{-1}$.
```

Step 3.9.4 Calculate the state estimate and update the covariance:

$$\hat{\boldsymbol{x}}^{(j)}{}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{(j)}(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}),$$
$$\boldsymbol{P}_k^{(j)} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k^{(j)} \boldsymbol{P}_{zz,k|k-1}^{(j)} \left(\boldsymbol{K}_k^{(j)}\right)^{\mathrm{T}}.$$

Step 3.9.5 Calculate the updated parameter of the IW distribution of measurement noise covariance:

$$\boldsymbol{X}_{i,k}^{(j)} = \boldsymbol{S}_k^{(j)} \xi_i + \hat{\boldsymbol{x}}_k^{(j)},$$
$$\boldsymbol{V}_k^{(j)} = \boldsymbol{V}_{k|k-1} + \tfrac{1}{2n}\sum_{i=1}^{2n}\left(\boldsymbol{z}_k - h(\boldsymbol{X}_{i,k}^{(j)},k)\right)\left(\boldsymbol{z}_k - h(\boldsymbol{X}_{i,k}^{(j)},k)\right)^{\mathrm{T}}.$$

Step 3.10 Until j = M, output $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}^{(M)}$, $\boldsymbol{P}_{k|k} = \boldsymbol{P}_k^{(M)}$, and $\boldsymbol{V}_k = \boldsymbol{V}_k^{(M)}$.

4. Numerical Simulation

To evaluate the performance of the proposed methods, the remote distance passive target tracking scenario [27] is considered. The motion model of the target is the coordinated turn, analyzing the performance changes of the four continuous–discrete robust algorithms proposed in this paper for data with Gaussian noise (Section 4.1), non-Gaussian noise (Section 4.2), or corrupted by outliers (Section 4.3). When considering random interference, coordinated turn motion is a typical nonlinear motion. The continuous-time motion model can be expressed by stochastic differential equation.

The state vector of the target is $\boldsymbol{x}(t) = \begin{bmatrix} x(t) & \dot{x}(t) & y(t) & \dot{y}(t) & \omega \end{bmatrix}^{\mathrm{T}}$; $x(t)$ and $y(t)$, $\dot{x}(t)$ and $\dot{y}(t)$ are the position and velocity of the target in the Cartesian coordinate system, respectively; and ω is a constant turn rate, which is 0.01 rad/s. The motion of single observer in the CA model with the initial state is $[0\text{km}, -0.025\text{km/s}, 0\text{km}, 0.02\text{km/s}]^{\mathrm{T}}$. The state equation of the system is $\boldsymbol{F}(\boldsymbol{x}(t)) = [\dot{x}(t), -\omega\dot{y}(t), \dot{y}(t), \omega\dot{x}(t), 0]^{\mathrm{T}}$. The random noise term is $\boldsymbol{w}(t) = [w_1(t), w_2(t), w_3(t), w_4(t), w_5(t)]^{\mathrm{T}}$, where its increment and state are independent of each other. Initial covariance matrix $\boldsymbol{Q} = \mathrm{diag}[0 \quad \tau_1 \quad 0 \quad \tau_1 \quad \tau_2]$, where $\tau_1 = \sqrt{2}$ and $\tau_2 = 7 \times 10^{-6}$. The initial state $\boldsymbol{x}(t_0) = [40\text{km}, 0\text{km/s}, 50\text{km}, 0.2\text{km/s}, 0.01\text{rad/s}]^{\mathrm{T}}$, and the initial covariance matrix $\boldsymbol{P}(t_0) = \mathrm{diag}[0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0]$. The measurement sampling interval is 1 min. The number of Monte Carlo simulations is 200. The experiment is based on the MATLAB simulation software with an Intel(R) Core (TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz.

Among them, the critical value of abnormal error discrimination in the HRSRCD-CKF algorithm is c = 1.345, the α quantile of the chi-square distribution in the MRSRCD-CKF algorithm is $\alpha_1 = 0.1, \alpha_2 = 0.1\%$, the Gaussian kernel bandwidth σ in the RSRCD-MCCKF algorithm is selected according to the simulation scenarios, the initial parameters in the VBSRCD-CKF algorithm are $v_0 = 600$, $\boldsymbol{V}_0 = 0.01$, $\rho = 1 - \exp(-4)$, and the number of internal loops of the algorithm is M = 10.

The discrete-time nonlinear measurement model is

$$\boldsymbol{z}_k = \left[\tan^{-1}(y_r(t_k)/x_r(t_k))\right] + \boldsymbol{r}_k, \tag{38}$$

where $x_r(t_k)$ and $y_r(t_k)$ are the relative position of the target and the observer at time k, $\boldsymbol{r}_k$ is the measurement noise, and the specific expression form is defined according to the simulation scenarios.

In order to evaluate the performance of filtering, the Root Mean Square Error (RMSE) of the position state and the velocity state are respectively defined as

$$\mathrm{RMSE}_{\mathrm{pos}}(t_k) = \left(\frac{1}{N}\sum_{i=1}^{N}\left\|x(t_k) - \hat{x}^i(t_k)\right\|_2^2\right)^{1/2}, \tag{39}$$

$$\mathrm{RMSE}_{\mathrm{vel}}(t_k) = \left(\frac{1}{N}\sum_{i=1}^{N}\left\|\dot{x}(t_k) - \hat{\dot{x}}^i(t_k)\right\|_2^2\right)^{1/2}, \tag{40}$$

where N is the number of Monte Carlo simulations, $\hat{x}^i(t_k)$ and $\hat{\dot{x}}^i(t_k)$ are the results of the i-th Monte Carlo state estimation, and $x(t_k)$, $\dot{x}(t_k)$ are the true position state and velocity state of the target, respectively. The trajectory of target and observer is shown in Figure 1.

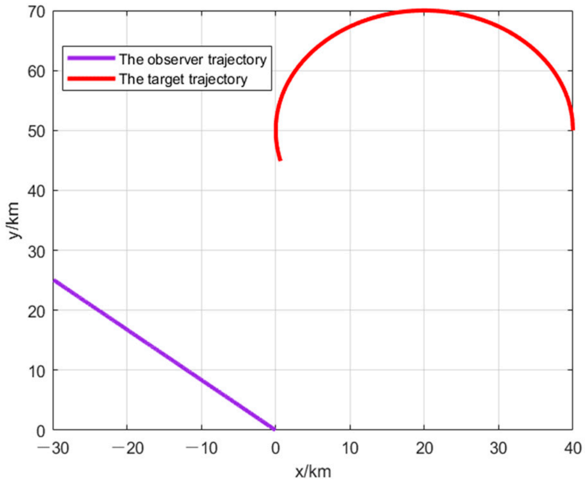


Figure 1. The trajectory of target and observer.

As shown in Figure 1, the bearings-only tracking scenario is tracking a two-dimensional target that is moving in a coordinated turn at a constant turn rate. The observer is maneuvered to achieve observability. In addition, it should be noted that the target tracked in this paper is the interference source on the ground, and the purpose of this paper is to improve the accuracy and robustness of this tracking process in different measurement environments through the continuous–discrete robust method.

4.1. Gaussian Noise

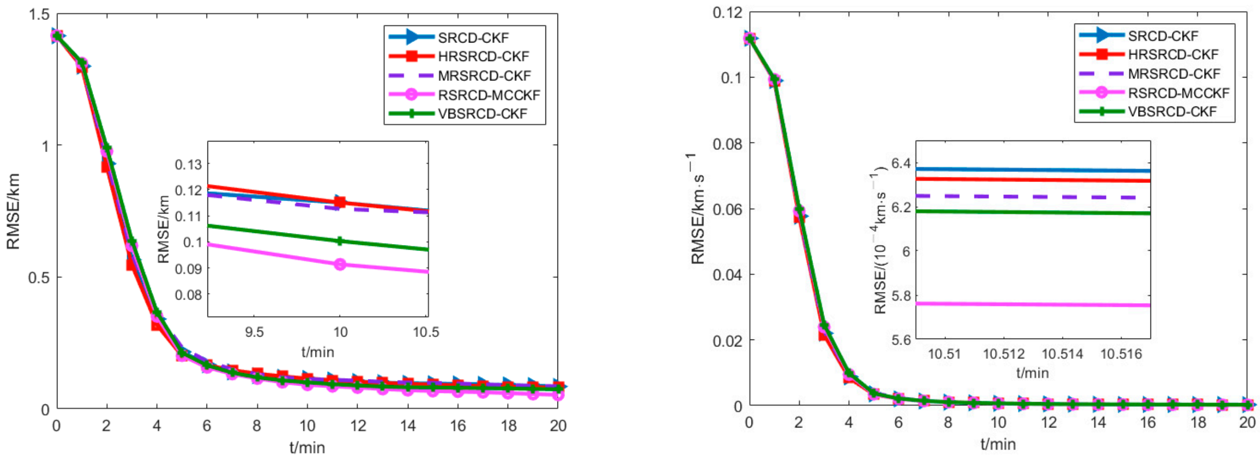
The measurement noise $\boldsymbol{r}_k$ satisfies the Gaussian distribution with zero mean, that is, $\boldsymbol{r}_k \sim N(\boldsymbol{0}, \boldsymbol{R}_k)$. The initial value of $\boldsymbol{R}_k$ is a one-dimensional constant value of 0.001 rad. Before comparing the proposed algorithms, the influence of Gaussian kernel bandwidth σ on the algorithm RSRCD-MCCKF is discussed. The RMSE_{pos} and RMSE_{vel} of the algorithm at the final time instant are shown in Table 1.

Table 1. Algorithm accuracy under Gaussian noise when kernel bandwidth is different.

Algorithms	**RMSE_{pos} [km]**	**RMSE_{vel} [km·s^{-1}]**
RSRCD-MCCKF (σ = 0.1)	—	—
RSRCD-MCCKF (σ = 1)	0.0528	0.000205
RSRCD-MCCKF (σ = 2)	0.0449	0.000188
RSRCD-MCCKF (σ = 5)	0.0482	0.000179
RSRCD-MCCKF (σ = 10)	0.0645	0.000211
RSRCD-MCCKF (σ = 50)	0.0771	0.000239
SRCD-CKF	0.0784	0.000247

As shown in Table 1, when the Gaussian kernel bandwidth σ changes, the algorithm shows different performance. When the value of σ is too small or too large, the filtering accuracy will be reduced. Among them, — refers to the filtering as diverged. When the value of σ increases gradually, its filtering performance has approached the SRCD-CKF algorithm, which is in line with the previous theoretical derivation. But under the premise of σ = 50, its filtering accuracy is still higher than the SRCD-CKF algorithm. In other words, as long as the proper value of σ is selected, the filtering accuracy of the RSRCD-MCCKF algorithm is higher than the traditional algorithm.

On this basis, the Gaussian kernel bandwidth σ = 5 is selected, and we can see the RMSE_{pos} and RMSE_{vel} comparison of each algorithm in Figure 2.



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

Figure 2. The RMSE comparison of each algorithm under Gaussian noise. (**a**) RMSE_{pos} of each algorithm under Gaussian noise; (**b**) RMSE_{vel} of each algorithm under Gaussian noise.

As shown in Figure 2, when the measurement noise is Gaussian noise, the previous trends of each algorithm are roughly the same, and they all tend to converge as the time increases. Among them, HRSRCD-CKF and MRSRCD-CKF have small fluctuations compared with the SRCD-CKF algorithm, and the accuracy is basically the same under normal measurement conditions. The accuracy of the VBSRCD-CKF algorithm is slightly higher than that of HRSRCD-CKF and MRSRCD-CKF; this reflects the higher accuracy of the variational Bayesian method in approximate estimation. The RSRCD-MCCKF has higher accuracy than the above algorithms, because the maximum correntropy criterion can utilize higher-order information of measurements, applying the Gaussian kernel to each element of the estimation error matrix, which has a better combination with SRCD-CKF.

4.2. Gaussian Mixture Noise

The measurement noise is Gaussian mixed noise. Gaussian mixed noise is a typical non-Gaussian distribution, $\boldsymbol{r}_k \sim 0.99N(\boldsymbol{0}, \boldsymbol{R}_k) + 0.01N(\boldsymbol{0{,}500} * \boldsymbol{R}_k)$. The RMSE_{pos} and RMSE_{vel} of the RSRCD-MCCKF algorithm at the final time instant, when the kernel bandwidth σ changes in the non-Gaussian noise environment, are shown in Table 2.

Table 2. Algorithm accuracy under non-Gaussian noise when kernel bandwidth is different.

Algorithms	**RMSE_{pos} [km]**	**RMSE_{vel} [km·s^{-1}]**
RSRCD-MCCKF (σ = 2)	—	0.0349
RSRCD-MCCKF (σ = 3)	1.0829	0.0043
RSRCD-MCCKF (σ = 5)	0.4716	0.0014
RSRCD-MCCKF (σ = 7)	0.4813	0.0014
RSRCD-MCCKF (σ = 10)	0.5375	0.0016
SRCD-CKF	0.6679	0.0019

As shown in Table 2, the value of σ that can suppress the non-Gaussian noise has changed. Compared with the Gaussian conditions, the filtering accuracy of the algorithms is reduced, but the appropriate σ can still suppress the non-Gaussian noise that appears in the measurement. Therefore, it is more important to adaptively select the value of σ according to different scenarios.

Here, the Gaussian kernel bandwidth σ = 5 is also selected under the condition of non-Gaussian noise. The RMSE_{pos} and RMSE_{vel} comparison of each algorithm is shown in Figure 3.

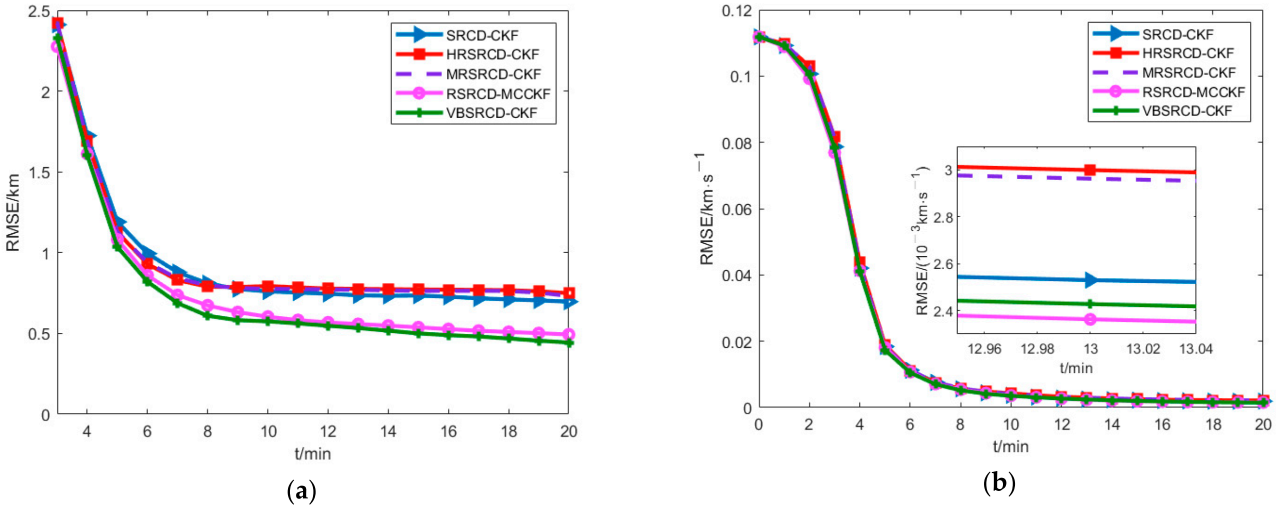


Figure 3. The RMSE comparison of each algorithm under non-Gaussian noise. (**a**) RMSE_{pos} of each algorithm under non-Gaussian noise; (**b**) RMSE_{vel} of each algorithm under non-Gaussian noise.

It can be seen from Figure 3 that when the measurement noise is non-Gaussian noise, the errors of each algorithm increase slightly. The HRSRCD-CKF algorithm and the MRSRCD-CKF algorithm cannot effectively suppress the non-Gaussian measurement noise, and the algorithm basically fails. The VBSRCD-CKF algorithm and the RSRCD-MCCKF algorithm can suppress non-Gaussian noise to a certain extent. The filtering performance of the two algorithms is basically equivalent and finally shows relatively high convergence accuracy, which reflects their ability to deal with non-Gaussian noise problems.

4.3. Gaussian Noise Together with Shot Noise

The measurement noise satisfies the Gaussian distribution with zero mean, and the measurement covariance matrix changes suddenly to 30 times the original at t = 10 min, $\boldsymbol{R}_k^{10} = 30\boldsymbol{R}_k$. Considering that when $\sigma = 5$, the filtering performance in RSRCD-MCCKF is good regardless of Gaussian noise or non-Gaussian noise; consequently, we still choose $\sigma = 5$ here. The RMSE_{pos} and RMSE_{vel} of each algorithm are shown in Figure 4.

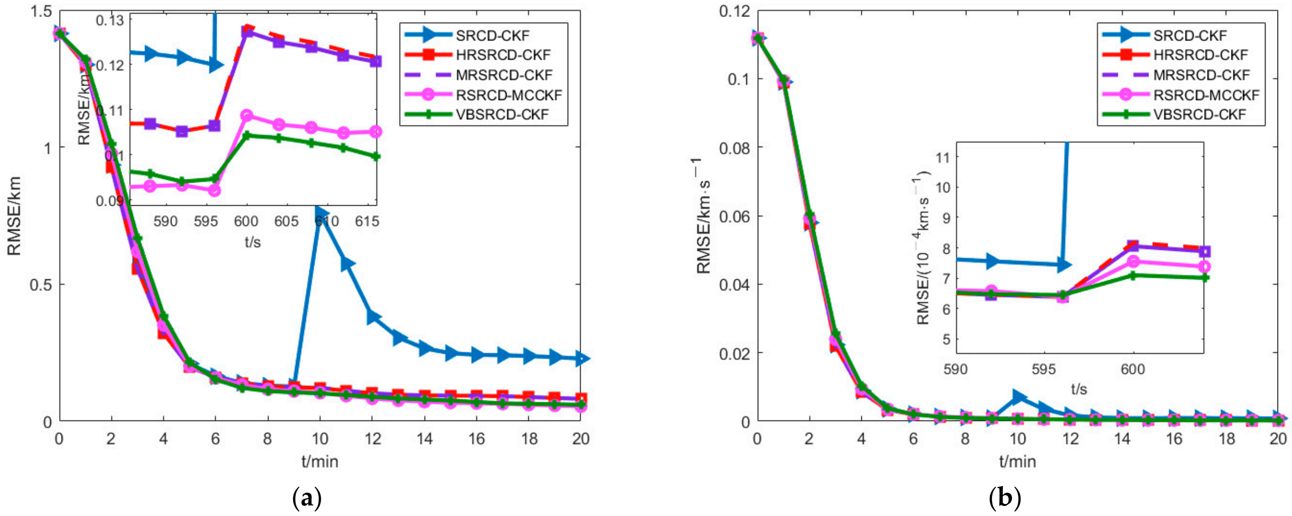


Figure 4. The RMSE comparison of each algorithm under abnormal measurement. (**a**) RMSE_{pos} of each algorithm under abnormal measurement; (**b**) RMSE_{vel} of each algorithm under abnormal measurement.

As shown in Figure 4, when the measurement is abnormal, the error of the SRCD-CKF algorithm increases abruptly, which is not robust. While the other algorithms effectively resist the abnormal measurement. Among them, the VBSRCD-CKF algorithm shows better robustness, and it has the most obvious effect on abnormal measurement suppression; the RSRCD-MCCKF algorithm is second only to the VBSRCD-CKF algorithm and shows good robustness with high accuracy, while the suppression effect of the HRSRCD-CKF algorithm and the MRSRCD-CKF algorithm is slightly worse, indicating that the RSRCD-MCCKF algorithm and the VBSRCD-MCCKF algorithm are more effective in dealing with abnormal measurement problems.

4.4. Computational Complexity

When comparing the computational complexity of each algorithm, the parameter settings are the same as in Sections 4.1 and 4.3, respectively. The relative computation time of SRCD-CKF is set to "1". The relative computation time of each algorithm is shown in Table 3.

Table 3. The relative computation time of each algorithm.

Algorithms	**Computation Time (Scenario 4.1)**	**Computation Time (Scenario 4.3)**
SRCD-CKF	1	1
HRSRCD-CKF	1.029	1.034
MRSRCD-CKF	1.005	1.018
RSRCD-MCCKF	1.076	1.080
VBSRCD-CKF	1.068	1.071

As shown in Table 3, the four robust algorithms all take longer than the initial SRCD-CKF algorithm, but their difference is not obvious. In addition, it can be observed that the computation time of each algorithm increases when interference is encountered, but the computation time has not increased much. Among them, the HRSRCD-CKF algorithm and the MRSRCD-CKF algorithm are relatively low in complexity. The computation time of the VBSRCD-CKF algorithm increases due to a certain number of inner loops. The computation time of the RSRCD-MCCKF is the longest because the calculation process includes more matrix inversion operations and exponential operations of Gaussian kernel function. However, considering the accuracy of the algorithm and application scenarios, it is acceptable within a similar time period.

Furthermore, it can be seen that the computation time of the proposed algorithms are longer than initial algorithm, but its improvement in accuracy and robustness is critical, which is the focus of this research. At the same time, considering that the increase in computing time is not obvious, the proposed algorithms can be used for other filtering problems.

5. Conclusions

In this paper, in order to evaluate the performance of robust methods in continuous–discrete tracking systems, four robust square-root continuous–discrete cubature Kalman filter algorithms are proposed. From the results, we can draw the following conclusions:

(1) The effectiveness of the algorithms in this paper is evaluated through a remote distance passive target tracking scenario, and the simulation results demonstrate the better environmental adaptability of the proposed algorithms in solving continuous-discrete robust problems. What's more, this paper aims to solve the slow tracking problem in remote distance passive target tracking, and the proposed algorithms can also be used in other navigation domains.

(2) As a common tool for data processing, Huber's estimation has a wide range of applications. The RSRCD-MCCKF algorithm and the VBSRCD-CKF algorithm proposed in this paper are better than the Huber's estimator, which provides an alternative means for the field of robust estimation.

(3) In the simulation results, we can see that the filtering performance of the algorithm without robustness is very poor. Therefore, when the system's data are contaminated by outliers, the relevant robust estimation method is indispensable.

Author Contributions: Conceptualization, R.H. and S.C.; methodology, H.H.; software, H.H.; validation, H.W. and H.H.; formal analysis, H.W.; investigation, H.H. and S.C.; resources, R.H.; data curation, H.H. and R.H.; writing—original draft preparation, H.H.; writing—review and editing, H.W.; visualization, H.H.; supervision, S.C.; project administration, S.C.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number [62073337, 61703420], the National Postdoctoral Program for Innovative Talent grant number [BX20190260], the China Postdoctoral Science Foundation [2019M663998], the Natural Science Foundation of Shannxi Province grant number [2020JQ-479], and the Young Elite Scientists Sponsorship Program by SNAST [20190109].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. He, R.K.; Chen, S.X.; Wu, H.; Chen, K.; Liu, J. Adaptive Covariance Feedback Cubature Kalman Filtering for Continuous-Discrete Bearings-Only Tracking System. *IEEE Access* **2019**, *7*, 2686–2694. [CrossRef]
2. Li, L.Q.; Wang, X.L.; Liu, Z.X.; Xie, W.X. Auxiliary truncated unscented Kalman filter for bearings-only maneuvering target tracking. *Sensors* **2017**, *17*, 972. [CrossRef] [PubMed]
3. Crouse, D. Basic tracking using nonlinear continuous-time dynamic models. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 4–41. [CrossRef]
4. Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulation. *IEEE Trans. Signal Processing* **2010**, *58*, 4977–4993. [CrossRef]
5. Crouse, D.F. Cubature Kalman Filters for Continuous-Time Dynamic Models Part I: Solutions Discretizing the Langevin Equation. In Proceedings of the 2014 IEEE Radar Conference, Cincinnati, OH, USA, 19–23 May 2014.
6. Crouse, D.F. Cubature Kalman Filters for Continuous-Time Dynamic Models Part II: A Solution Based on Moment Matching. In Proceedings of the 2014 IEEE Radar Conference, Cincinnati, OH, USA, 19–23 May 2014.
7. Kulikov, G.Y.; Kulikova, M.V. Accurate Numerical Implementation of the Continuous-Discrete Extended Kalman Filter. *IEEE Trans. Autom. Control* **2014**, *59*, 273–279. [CrossRef]
8. Wang, J.L.; Zhang, D.X.; Shao, X.W. New version of continuous-discrete cubature Kalman filtering for nonlinear continuous-discrete systems. *ISA Trans.* **2019**, *91*, 174–783. [CrossRef]
9. Kulikov, G.Y.; Kulikova, M.V. NIRK-based Accurate Continuous-Discrete Extended Kalman filters for Estimating Continous-Time Stochastic Target Tracking Models. *J. Comput. Appl. Math.* **2016**, *316*, 260–270. [CrossRef]
10. He, R.K.; Chen, S.X.; Wu, H.; Chen, K. Stochastic feedback based continuous-discrete cubature Kalman filtering for bearings-only tracking. *Sensors* **2018**, *18*, 1959. [CrossRef]
11. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *Radar Signal Processing IEE Proc. Part F* **1993**, *140*, 107–113. [CrossRef]
12. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing* **2002**, *50*, 174–188. [CrossRef]
13. Meinhold, R.J.; Singpurwalla, N.D. Robustification of Kalman filter methods. *J. Am. Stat. Assoc.* **1989**, *84*, 479–486. [CrossRef]
14. Huber, P.J.; Ronchetti, E.M. *Robust Statistics*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2009.
15. Jureckova, J.; Picek, J.; Schindler, M. *Robust Statistical Methods with R*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2019.
16. Chang, J.B.; Liu, M. M-estimator-based robust Kalman filter for systems with process modeling errors and rank deficient measurement models. *Nonlinear Dyn.* **2015**, *80*, 1431–1449. [CrossRef]
17. Wu, H.; Chen, S.X.; Chen, K.; Yang, B.F. Robust cubature Kalman filter target tracking algorithm based on genernalized M-estiamtion. *Acta Phys. Sin.* **2015**, *64*, 456–463. [CrossRef]
18. Wu, H.; Chen, S.X.; Chen, K.; Yang, B.F. Robust Derivative-Free Cubature Kalman Filter for Bearings-Only Tracking. *J. Guid. Control. Dyn.* **2016**, *39*, 1865–1870. [CrossRef]
19. Särkkä, S.; Nummenmaa, A. Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Trans. Autom. Control.* **2009**, *54*, 596–600. [CrossRef]

20. Wang, G.Q.; Li, N.; Zhang, Y.G. Maximum correntropy unscented Kalman and information filters for non-Gaussian measurement noise. *J. Frankl. Inst.* **2017**, *354*, 8659–8677. [CrossRef]
21. Liu, X.; Qu, H.; Zhao, J.; Yue, P.; Wang, M. Maximum correntropy unscented Kalman filter for spacecraft relative state estimation. *Sensors* **2016**, *16*, 1530. [CrossRef]
22. Liu, X.; Chen, B.; Xu, B.; Wu, Z.; Honeine, P. Maximum correntropy unscented filter. *Int. J. Syst. Sci.* **2017**, *48*, 1607–1615. [CrossRef]
23. Särkkä, S.; Solin, A. On continuous-discrete cubature Kalman filtering. *IFAC Proc. Vol.* **2012**, *45*, 1221–1226. [CrossRef]
24. Särkkä, S. On unsented Kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Trans. Autom. Control* **2007**, *52*, 1631–1641. [CrossRef]
25. Chen, B.D.; Liu, X.; Zhao, H.Q.; Principe, J.C. Maximum correntropy Kalman filter. *Automatica* **2017**, *76*, 70–77. [CrossRef]
26. Särkkä, S.; Hartikainen, J. Non-linear noise adaptive Kalman filtering via variational Bayes. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP) Conference, Southampton, UK, 22–25 September 2013.
27. He, R.K.; Chen, S.X.; Wu, H.; Zhang, F.Z.; Chen, K. Efficient extended cubature Kalman filtering for nonlinear target tracking. *Int. J. Syst. Sci.* **2020**, *52*, 392–406. [CrossRef]