

## Article

# Maximum Correntropy Criterion Kalman Filter for $\alpha$ -Jerk Tracking Model with Non-Gaussian Noise

Bowen Hou <sup>1</sup>, Zhangming He <sup>1,2</sup>, Xuanying Zhou <sup>1</sup>, Haiyin Zhou <sup>1</sup>, Dong Li <sup>3</sup>  
and Jiongqi Wang <sup>1,2,\*</sup>

<sup>1</sup> Department of Mathematics and System Science, National University of Defense Technology, Fuyuan Road No. 1, Changsha 410072, China; houbowen12@nudt.edu.cn (B.H.); hzmnudt@163.com (Z.H.); julia\_chow07@163.com (X.Z.); zhy\_gfkd@163.com (H.Z.)

<sup>2</sup> Beijing Institute of Control Engineering, China Academy of Space Technology, Beijing 100080, China

<sup>3</sup> Unit 94, PLA 91550, Dalian 116023, China; nudtli dong@163.com

\* Correspondence: wjq\_gfkd@163.com; Tel.: +86-137-8719-8870

Received: 15 November 2017; Accepted: 26 November 2017; Published: 29 November 2017

**Abstract:** As one of the most critical issues for target track,  $\alpha$ -jerk model is an effective maneuver target track model. Non-Gaussian noises always exist in the track process, which usually lead to inconsistency and divergence of the track filter. A novel Kalman filter is derived and applied on  $\alpha$ -jerk tracking model to handle non-Gaussian noise. The weighted least square solution is presented and the standard Kalman filter is deduced firstly. A novel Kalman filter with the weighted least square based on the maximum correntropy criterion is deduced. The robustness of the maximum correntropy criterion is also analyzed with the influence function and compared with the Huber-based filter, and, moreover, the kernel size of Gaussian kernel plays an important role in the filter algorithm. A new adaptive kernel method is proposed in this paper to adjust the parameter in real time. Finally, simulation results indicate the validity and the efficiency of the proposed filter. The comparison study shows that the proposed filter can significantly reduce the noise influence for  $\alpha$ -jerk model.

**Keywords:** Kalman filter;  $\alpha$ -jerk model; maximum correntropy criterion; non-Gaussian noise; robustness; influence function; kernel size

## 1. Introduction

Target track is a real-time estimation process to obtain the position, the velocity and other motion parameters for the tracked target through the measurement data. Because of the uncertainty and nonlinearity of the maneuver target motion process and the target measurement process, a lot of algorithms were researched and proposed for the maneuver target track in the past [1,2]. To sum up, the track algorithms study has been focused on two parts, i.e., maneuver target modeling and nonlinear filter design.

For the maneuver target modeling, many models were proposed to describe the target maneuvers. Singer model [3] is a classic model proposed by Singer in 1970. In 1982, Bar-shalom and Birmiwal [4] proposed a maneuver target track model with the maneuver detection. In 1984, Zhou and Kumar [5] proposed the current statistical model based on the Singer model. The jerk model proposed by Mehrotra [6] was another valid method to track the maneuver target. Among the models above, jerk model has the highest order. A higher model order can present higher mobility of target. Jerk model broadens the adaptive target maneuver form and is more suitable for high maneuver targets [5]. However, for the traditional jerk model, “jerk”  $\alpha$  should be predefined and may be time-varying in the practical process. Therefore, Luo proposed  $\alpha$ -jerk model [7], which considered  $\alpha$  as a parameter to be estimated by the expanding-dimension method in the real-time filter process.

For the nonlinear filter design, a lot of techniques were proposed according to the different optimization rules, including the function approximation, the posteriori probability density approximation and the stochastic model approximation [8–11]. Extended Kalman Filter (EKF) is a typical technique of the function approximation based on Taylor series expansion. Unscented Kalman Filter (UKF) is the most common way of the sampling-based moment approximation [12,13]. The stochastic model approximation [14] can approximate the nonlinear function in some probabilistic senses. However, it has not gained popularity mainly because the expectations involved are not easy to evaluate. No matter what nonlinear filter methods are used, they are applied on the state estimation with Gaussian white noises in most time.

However, in practical applications, especially in the maneuver target track system, non-Gaussian noises do exist and have substantial effects on the state estimation. Up to now, some practical models are proposed, such as Gaussian mixture noise model [15], glint noise model [16] and so on.

In a non-Gaussian process, the performance of many nonlinear filters can break down [17]. However, the filter robustness against non-Gaussian noises can be improved by the following approaches:

1. The first approach is to develop filters for the systems with non-Gaussian noises directly. Noise distributions such as heavy-tailed distributions and  $t$ -distributions are considered in these filters [18,19]. However, it is difficult to handle more than one dimension, which limits its applicability [20].
2. Approximating the posteriori probability density is another practical approach to handle the non-Gaussian noises. The unscented Kalman filter (UKF) uses the unscented transformation (UT) technique to capture the mean and the covariance of the state estimation with sigma points [21]. The ensemble Kalman filter (EnKF) is a method to approximate the state estimation with a set of samples to handle non-Gaussian noises [22]. Gaussian sum filter (GSF) is an algorithm to obtain the filtering distribution and the predictive distribution recursively approximated as Gaussian mixtures [23–25].
3. Huber-based robust filter is a classical method to handle the non-Gaussian noises and outliers [26]. It has become the research focus these past several years and been applied in the field of the navigation and the target track [27–30].
4. A new robust Kalman filter is proposed by Chang [31] in recent years. It handles the outliers based on the hypothesis testing theory, which defines a judging index as the square of the Mahalanobis distance from the observation to its prediction. It can effectively resist the heavy-tailed distribution of the observation noises and the outliers in the actual observations.
5. Multi-sensor data fusion Kalman filter is a fuzzy logical method proposed by Rodger [32]. It can effectively improve the computational burden and the robustness of Kalman filter. Furthermore, it has been applied on the vehicle health maintenance system.
6. Maximum correntropy criterion is the latest optimization criterion that is used for improving Kalman filter. Maximum correntropy Kalman filter (MCKF) is a newly proposed filter to process the non-Gaussian noises [33]. In addition, several improved MCKF algorithms have been proposed and applied on state estimation [34–37].

Maximum correntropy criterion Kalman filter (MCCKF) [20] is the latest approach to handle the non-Gaussian noises with Gaussian kernel. It is different from the MCKF in the form and the derivation process. The aforementioned MCCKF has been proved to have higher accuracy than other approaches in a benchmark navigation problem with the non-Gaussian noises. However, it does not have an optimal rule for the kernel size selection, which is important for the robustness of MCCKF. Furthermore, it has not been applied on the maneuver target track. Thus, we adopt the Kalman filter based on the maximum correntropy criterion for  $\alpha$ -jerk model with the non-Gaussian process to realize the maneuver target track.

The key contributions of this paper are as follows. Firstly, the robustness of Kalman filter algorithm based on the maximum correntropy criterion is introduced in the presence of the non-Gaussian noises. Then, the influence function of the filter algorithm, for the first time, is analyzed in this paper. Moreover,

an optimal kernel size is obtained by a novel adaptive size adjusting skill. Finally, the large outliers and the Gaussian mixture noises in the maneuver target track with  $\alpha$ -jerk model are properly handled by MCCKF.

The organization of this paper proceeds as follows. The  $\alpha$ -jerk model is briefly reviewed in Section 2. In Section 3, we design the algorithm steps for MCCKF, analyze the robustness performance with the influence function and propose a novel kernel size selection method. In Section 4, simulation results are presented to compare the performances of different filter algorithms with respect to different noise conditions. Finally, conclusions are drawn in Section 5.

## 2. $\alpha$ -Jerk Model

Jerk model is a classical target tracking model. “Jerk” means the acceleration rate [6]. It is described as follows:

$$\dot{j}(t) = -\alpha j(t) + \omega(t), \quad (1)$$

where  $j(t) = \ddot{x}(t)$  is the target “jerk”,  $x(t)$  is the target position,  $\alpha$  is the correlation parameter permitting the modeling of different classes of targets,  $\omega(t)$  is the process noises with the zero mean and the variance  $\sigma_\omega^2 = 2\alpha\sigma_j^2$ , and  $\sigma_j^2$  is the variance for the target “jerk”. In the model,  $\alpha$  is a fixed value. However, it is difficult to predefine and may be time-varying in the practical process.

$\alpha$ -jerk model considers  $\alpha$  as a parameter to be estimated and supposes that  $\alpha$  is a state variable. In this way, it will be real-time estimated in the filter process. Consider the following equation:

$$\dot{\alpha}(t) = \varepsilon(t), \quad (2)$$

where  $\varepsilon(t)$  denotes the zero mean Gaussian noises with the variance  $\sigma_\varepsilon^2$ .

Combined with Equation (2) and Jerk model [6], the state equation for the continuous time  $\alpha$ -jerk when only one variable,  $x$ , in the state variable vector  $\begin{bmatrix} x & \dot{x} & \ddot{x} & \alpha \end{bmatrix}^T$ , is considered, the  $\alpha$ -jerk model can be written as

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega(t) \\ \varepsilon(t) \end{bmatrix}, \quad (3)$$

where  $x, \dot{x}, \ddot{x}$  represent the position, the velocity, the acceleration and the jerk of the aerial target in the  $x$ -direction, respectively. Denote

$$\begin{cases} \mathbf{X} = \begin{bmatrix} x & \dot{x} & \ddot{x} & \alpha \end{bmatrix}^T \\ \mathbf{W}(t) = \begin{bmatrix} 0 & 0 & 0 & \omega(t) & \varepsilon(t) \end{bmatrix}^T \end{cases} \quad (4)$$

and

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

The state equation can be expressed as:

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{W}(t). \quad (6)$$

Equation (6) is discretized as:

$$\mathbf{X}_k = \Phi_{k|k-1}\mathbf{X}_{k-1} + \mathbf{W}_k, \quad (7)$$

where

$$\Phi_{k|k-1} = \exp(AT), \quad (8)$$

where  $T$  is the time interval and  $\Phi_{k|k-1}$  is the state transition matrix;  $\mathbf{W}_k$  is the process noises at time  $k$ .

From Equation (8), the discrete state transition matrix can be written as:

$$\Phi_{k|k-1} = \begin{bmatrix} 1 & T & T^2/2 & p_1 & 0 \\ 0 & 1 & T & q_1 & 0 \\ 0 & 0 & 1 & r_1 & 0 \\ 0 & 0 & 0 & s_1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where

$$\begin{aligned} p_1 &= (2 - 2\alpha T + \alpha^2 T^2 - 2e^{-\alpha T}) / (2\alpha^3) \\ q_1 &= (e^{-\alpha T} - 1 + \alpha T) / \alpha^2 \\ r_1 &= (1 - e^{-\alpha T}) / \alpha \\ s_1 &= e^{-\alpha T} \end{aligned}. \quad (10)$$

The variance of the process noises  $\mathbf{W}_k$  is

$$\mathbf{Q}(k) = \begin{bmatrix} q_{11}\sigma_\omega^2 & q_{12}\sigma_\omega^2 & q_{13}\sigma_\omega^2 & q_{14}\sigma_\omega^2 & 0 \\ q_{21}\sigma_\omega^2 & q_{22}\sigma_\omega^2 & q_{23}\sigma_\omega^2 & q_{24}\sigma_\omega^2 & 0 \\ q_{31}\sigma_\omega^2 & q_{32}\sigma_\omega^2 & q_{33}\sigma_\omega^2 & q_{34}\sigma_\omega^2 & 0 \\ q_{41}\sigma_\omega^2 & q_{42}\sigma_\omega^2 & q_{43}\sigma_\omega^2 & q_{44}\sigma_\omega^2 & 0 \\ 0 & 0 & 0 & 0 & T\sigma_\varepsilon^2 \end{bmatrix}, \quad (11)$$

where

$$\begin{aligned} q_{11} &= (1/2\alpha^7) (\alpha^5 T^5/10 - \alpha^4 T^4/2 + 4\alpha^3 T^3/3 \\ &\quad + 2\alpha T - 2\alpha^2 T^2 - 3 + 4e^{-\alpha T} + 2\alpha^2 T^2 e^{-\alpha T} - e^{-2\alpha T}) \\ q_{12} &= q_{21} = (1/2\alpha^6) (1 - 2\alpha T + 2\alpha^2 T^2 - \alpha^3 T^3 + \alpha^4 T^4/4 \\ &\quad + e^{-2\alpha T} + 2\alpha T e^{-\alpha T} - 2e^{-\alpha T} - \alpha^2 T^2 e^{-\alpha T}) \\ q_{13} &= q_{31} = (1/2\alpha^5) (\alpha^3 T^3/3 + 2\alpha T - \alpha^2 T^2 - 3 \\ &\quad + 4e^{-\alpha T} + \alpha^2 T^2 e^{-\alpha T} - e^{-2\alpha T}) \\ q_{14} &= q_{41} = (1/2\alpha^4) (1 - 2e^{-\alpha T} - \alpha^2 T^2 e^{-\alpha T} + e^{-2\alpha T}) \\ q_{22} &= (1/2\alpha^5) (1 - e^{-2\alpha T} + 2\alpha^3 T^3/3 + 2\alpha T - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T}) \\ q_{23} &= q_{32} = (1/2\alpha^4) (1 - 2\alpha T + \alpha^2 T^2 + 2\alpha T e^{-\alpha T} + e^{-2\alpha T} - 2e^{-\alpha T}) \\ q_{24} &= q_{42} = (1/2\alpha^3) (1 - 2\alpha T e^{-2\alpha T} - e^{-2\alpha T}) \\ q_{33} &= (1/2\alpha^3) (4e^{-\alpha T} + 2\alpha T - e^{-2\alpha T} - 3) \\ q_{34} &= q_{43} = (1/2\alpha^2) (1 - 2e^{-\alpha T} + e^{-2\alpha T}) \\ q_{44} &= (1/2\alpha) (1 - e^{-2\alpha T}). \end{aligned} \quad (12)$$

The measurement equation can be expressed as:

$$\mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{V}_k, \quad (13)$$

where  $\mathbf{H}_k$  is the measurement matrix at time  $k$ ;  $\mathbf{V}_k$  is the measurement noises and independent of the process noises  $\mathbf{W}_k$ .

In most situations,  $\mathbf{V}_k$  is the Gaussian white noises. However, in practice, due to the target glint, the scattering of the radar tracking equipment and the faults of the operators [16,38] and so on, the measurement noises may be non-Gaussian or some outliers exist in the measurement data. It can behave as:

$$\mathbf{V}_k = \mathbf{V}_k^{(1)} + \mathbf{V}_k^{(2)} \quad (14)$$

or

$$\mathbf{V}_k = \mathbf{V}_k^{(1)} + \Delta \mathbf{V}_k, \quad (15)$$

where  $\mathbf{V}_k^{(1)}$  denotes the Gaussian white noises,  $\mathbf{V}_k^{(2)}$  obeys another Gaussian distribution different from  $\mathbf{V}_k^{(1)}$ , and  $\Delta \mathbf{V}_k$  denotes the outliers.

In this paper, a new filter algorithm will be proposed to deal with the non-Gaussian noises like Equations (14) and (15).

### 3. Design for Maximum Correntropy Criterion Kalman Filter with Non-Gaussian Noise

#### 3.1. Standard Kalman Filter

From Section 2,  $\alpha$ -jerk model for the maneuver target track can be expressed as:

$$\begin{cases} \mathbf{X}_k = \Phi_{k|k-1} \mathbf{X}_{k-1} + \mathbf{W}_k, \\ \mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{V}_k. \end{cases} \quad (16)$$

Assume  $\mathbf{W}_k$  and  $\mathbf{V}_k$  are independent Gaussian white noises, satisfying

$$\mathbf{E}(\mathbf{W}_k \mathbf{W}_j^T) = \begin{cases} \mathbf{Q}_k, & k = j, \\ 0, & k \neq j, \end{cases}, \quad \mathbf{E}(\mathbf{V}_k \mathbf{V}_j^T) = \begin{cases} \mathbf{R}_k, & k = j, \\ 0, & k \neq j, \end{cases} \quad (17)$$

where  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are known positive definite matrix.

In the KF, the predicted state  $\hat{\mathbf{X}}_{k|k-1}$  satisfies

$$\mathbf{E} \left[ \left( \mathbf{X}_k - \hat{\mathbf{X}}_{k|k-1} \right) \left( \mathbf{X}_k - \hat{\mathbf{X}}_{k|k-1} \right)^T \right] = \mathbf{P}_{k|k-1}, \quad (18)$$

where  $\mathbf{P}_{k|k-1}$  is the predicted covariance. One very reasonable estimate  $\mathbf{X}_k$ , taking into account the measurement  $\mathbf{Z}_k$ , is the weighted least squares estimate, which we call  $\hat{\mathbf{X}}_k$ . For this estimate, we choose  $\hat{\mathbf{X}}_k$  as the value of  $\mathbf{X}_k$  that minimizes the quadratic form

$$J = \frac{1}{2} \left[ \left( \mathbf{X}_k - \hat{\mathbf{X}}_{k|k-1} \right)^T \mathbf{P}_{k|k-1}^{-1} \left( \mathbf{X}_k - \hat{\mathbf{X}}_{k|k-1} \right) + \left( \mathbf{Z}_k - \mathbf{H}_k \mathbf{X}_k \right)^T \mathbf{R}_k^{-1} \left( \mathbf{Z}_k - \mathbf{H}_k \mathbf{X}_k \right) \right]. \quad (19)$$

Note that the weighting matrices,  $P_{k|k-1}^{-1}$  and  $R_k^{-1}$ , are the inverse matrices of the prior expectations of  $(X_k - \hat{X}_{k|k-1})(X_k - \hat{X}_{k|k-1})^T$  and  $(Z_k - H_k X_k)(Z_k - H_k X_k)^T$ , respectively, and the predicted state is obtained as follows:

$$\hat{X}_{k|k-1} = \Phi_{k|k-1} X_{k-1}. \quad (20)$$

The value  $\hat{X}_k = \arg \min_{X_k} J(X_k)$  will be the best estimate. The detailed proof is given in [39] to show that Kalman filter can be deduced by Equation (19).

Therefore, Kalman filter can be deduced by solving

$$\frac{\partial J}{\partial \hat{X}_k} = 0. \quad (21)$$

Therefore, Kalman filter equations are expressed as:

$$\begin{aligned} \hat{X}_{k|k-1} &= \Phi_{k|k-1} \hat{X}_{k-1} \\ P_{k|k-1} &= \Phi_{k|k-1} P_{k-1|k-1} \Phi_{k|k-1}^T + Q_k \\ K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1}^{-1} H_k^T + R_k^{-1})^{-1} \\ \hat{X}_k &= \Phi_{k|k-1} \hat{X}_{k|k-1} + K_k (Z_k - H_k \hat{X}_{k|k-1}) \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}. \end{aligned} \quad (22)$$

The standard Kalman filter has been applied in many fields. However, it is under the assumptions that the process noises and the measurement noises are both Gaussian white noises. Therefore, it cannot have a good performance in a non-Gaussian condition, which is introduced in Section 2. Thus, some robust Kalman filtering algorithms were developed [40]. In this paper, a latest robust Kalman filter, maximum correntropy criterion Kalman filter, is deduced based on correntropy theory. It has higher accuracy than some other approaches in the presence of non-Gaussian noises.

### 3.2. Design of Maximum Correntropy Criterion Kalman Filter

#### 3.2.1. Correntropy

Correntropy [41] defines a metric in the data space and it is directly related to information entropy. It induces a new metric in the sample space that is equivalent to the 2-norm distance [42–44]. Given two random variables,  $A, B \in \mathbb{R}$  with the joint probability density function (PDF)  $p(a, b)$ . Then, the definition of the correntropy between two random variables is as follows:

**Definition 1.** Correntropy is a generalized similarity measure between two arbitrary scalar random variables  $A$  and  $B$  defined by

$$C_\lambda(A, B) = E[\kappa_\lambda(A, B)] = \iint \kappa_\lambda(a, b) p(a, b) da db, \quad (23)$$

where  $E$  is the expectation and  $\kappa_\lambda(\cdot, \cdot)$  is any continuous positive definite kernel function,  $\lambda > 0$  is the kernel size.

In this paper, the kernel function is chosen as the Gaussian kernel given by

$$\kappa_\lambda(a, b) = G_\lambda(e) = \exp\left(-\frac{e^2}{2\lambda^2}\right), \quad (24)$$

where  $e = a - b$ .

In practice, the joint PDF  $p(a, b)$  is unknown and a finite number of sample data are available. Therefore, the correntropy can be estimated by the sample mean estimator:

$$\hat{C}_\lambda(A, B) = \frac{1}{N} \sum_{i=1}^N G_\lambda(e_i), \quad (25)$$

where  $e_i = a_i - b_i$ , with  $\{a_i, b_i\}_{i=1}^N$  being  $N$  samples drawn from  $p(a, b)$ .

When  $\lambda = 1$ ,  $a_i \in [0, 10]$ ,  $b_i \in [0, 10]$  and the samples number is 50, the correntropy can be shown as Figure 1:

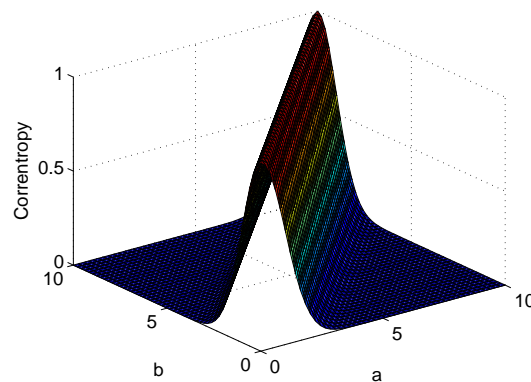


Figure 1. Correntropy when  $\lambda = 1$ .

As can be seen from Figure 1, the correntropy can be used as a similarity measure in the joint space. In addition, when  $a_i = b_i$ , the correntropy reaches the maximum value. The cost function that incarnates the maximum correntropy criterion (MCC) is:

$$J_{MCC} = \max \sum_{i=1}^N G_\lambda(e_i). \quad (26)$$

MCC has a probabilistic meaning of maximizing the error probability density at the origin [41], and it has the advantages that it is a local criterion of similarity and it is very useful for non-Gaussian noise [45]. In other words, compared with the mean square error (MSE), MCC is local along the bisector of the joint space, which means that the maximum correntropy value is mainly decided by the kernel function along the line  $A = B$  [46].

### 3.2.2. Maximum Correntropy Criterion Kalman Filter Algorithm

Similar to Equation (19), the maximum correntropy criterion Kalman filter (MCCKF) is designed based on the MCC theory. We have the correntropy-based objective function:

$$\hat{X}_k = \arg \max_{X_k} J(X_k), \quad (27)$$

where

$$J(X_k) = G_\lambda \left( \left\| X_k - \Phi_{k|k-1} \hat{X}_{k-1} \right\|_{P_{k|k-1}^{-1}} \right) + G_\lambda \left( \left\| Z_k - H_k X_k \right\|_{R_k^{-1}} \right), \quad (28)$$

where  $R_k^{-1}$  and  $P_{k|k-1}^{-1}$  are chosen as the weighting matrices.

The derivative of Equation (28) with respect to  $\mathbf{X}_k$  is:

$$\begin{aligned} G_\lambda \left( \left\| \mathbf{X}_k - \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1} \right\|_{P_{k|k-1}^{-1}} \right) P_{k|k-1}^{-1} (\mathbf{X}_k - \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1}) \\ - G_\lambda \left( \left\| \mathbf{Z}_k - \mathbf{H}_k \mathbf{X}_k \right\|_{R_k^{-1}} \right) \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \mathbf{X}_k) = 0. \end{aligned} \quad (29)$$

Therefore,  $\hat{\mathbf{X}}_k$  is the solution of Equation (29). Let

$$P_{k|k-1}^{-1} (\hat{\mathbf{X}}_k - \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1}) = L_k \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k), \quad (30)$$

where  $L_k = \frac{G_\lambda \left( \left\| \mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k \right\|_{R_k^{-1}} \right)}{G_\lambda \left( \left\| \hat{\mathbf{X}}_k - \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1} \right\|_{P_{k|k-1}^{-1}} \right)}$ . Then,  $(L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1})$  is added and subtracted on the right-hand side of Equation (30) to obtain

$$\begin{aligned} P_{k|k-1}^{-1} (\hat{\mathbf{X}}_k - \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1}) = \\ L_k \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k) + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1} - L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}, \end{aligned} \quad (31)$$

$$\begin{aligned} P_{k|k-1}^{-1} \hat{\mathbf{X}}_k - P_{k|k-1}^{-1} \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1} &= P_{k|k-1}^{-1} \hat{\mathbf{X}}_k - P_{k|k-1}^{-1} \hat{\mathbf{X}}_{k|k-1} = \\ L_k \mathbf{H}_k^T R_k^{-1} \mathbf{Z}_k - L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_k & \\ + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1} - L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}, & \end{aligned} \quad (32)$$

$$\begin{aligned} (P_{k|k-1}^{-1} + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k) \hat{\mathbf{X}}_k \\ = P_{k|k-1}^{-1} \hat{\mathbf{X}}_{k|k-1} + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1} + L_k \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}) \\ = (P_{k|k-1}^{-1} + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k) \hat{\mathbf{X}}_{k|k-1} + L_k \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}), \end{aligned} \quad (33)$$

$$\begin{aligned} \hat{\mathbf{X}}_k &= \hat{\mathbf{X}}_{k|k-1} + (P_{k|k-1}^{-1} + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k)^{-1} L_k \mathbf{H}_k^T R_k^{-1} (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}) \\ &= \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}). \end{aligned} \quad (34)$$

In addition, then the MCCKF algorithm can be derived from Equation (34) as follows:

1. State Prediction

$$\hat{\mathbf{X}}_{k|k-1} = \Phi_{k|k-1} \hat{\mathbf{X}}_{k-1}, \quad (35)$$

2. Covariance Prediction

$$P_{k|k-1} = \Phi_{k|k-1} P_{k-1|k-1} \Phi_{k|k-1}^T + Q_k, \quad (36)$$

3. Filter Gain

$$\mathbf{K}_k = (P_{k|k-1}^{-1} + L_k \mathbf{H}_k^T R_k^{-1} \mathbf{H}_k)^{-1} L_k \mathbf{H}_k^T R_k^{-1}, \quad (37)$$

4. State Update

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k (\mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1}), \quad (38)$$



## 5. Covariance Update

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T. \quad (39)$$

### 3.3. Robustness of MCCKF

The advantage of MCCKF is that the algorithm uses a method similar to Huber-based filter (HF) [47], which can resist the non-Gaussian noises because the robust function of HF does not increase rapidly as the estimate residual grows. Next, the algorithm robustness is analyzed with the influence function [48] in theory and is compared with HF through the weighting function.

HF is achieved by introducing the concept of the robustness in the maximum likelihood as proposed by Huber [47]. Recall that the maximum likelihood estimator is from a data observation  $Z = \{z_1, \dots, z_N\}$ , a state variable  $X = \{x_1, \dots, x_N\}$ . Assuming that  $e = Z - HX$  obeys the distribution and  $p(e; X)$  is known except for the parameters  $X$ , the MLE of  $X$  can be written as

$$\hat{X}_{MLE} = \arg \max_X p(e_1, \dots, e_N; X). \quad (40)$$

If  $e_1, \dots, e_N$  are i.i.d. observations, it can be rewritten as

$$\begin{aligned} \hat{X}_{MLE} &= \arg \max_X \prod_{i=1}^N p(e_i; X) = \arg \max_X \sum_{i=1}^N \ln p(e_i; X) \\ &= \arg \min_X \sum_{i=1}^N (-\ln p(e_i; X)). \end{aligned} \quad (41)$$

Assume that  $\rho$  is a function that satisfies  $\rho(e_i; X) = a(-\ln p(e_i; X)) + b$ , where  $a > 0$  and  $b$  is irrelevant to  $X$ . Therefore,  $\hat{X}_{MLE}$  satisfies

$$\sum_{i=1}^N \rho(e_i; X) = \sum_{i=1}^N \rho(z_i - Hx_i) = \min_X \sum_{i=1}^N \rho(e_i; X). \quad (42)$$

For some parametric models, if the errors in the target track model are assumed as purely stochastic white and follow a Gaussian distribution with zero mean and constant variance  $\sigma^2$ , then the maximum likelihood estimation can be solved. However, in practice, the measurement equipment are affected by external environment or break down. Therefore, the measurement outliers cannot be avoided. In addition, some errors obey the Gaussian distributions with different mean values and variances with most errors. For example, for one error, its distribution can be described as follows:

$$F(e_i; X) = (1 - v) F_0(e_i; X) + v F_c(e_i; X), \quad (43)$$

where  $F(\cdot)$  denotes the mixture distribution,  $F_0(\cdot)$  is a known normal distribution,  $F_c(\cdot)$  is a contaminated distribution that is unknown,  $v$  denotes the contamination ratio [49], which is usually much smaller than 1. For Equation (43), it is impractical to solve the maximum likelihood estimation by Equation (42).

For this problem, Huber promoted that  $\rho$  can change freely within limits. The method to solve Equation (42) is called  $M$ -estimation.

Huber defined the  $M$ -estimators as a generalization of maximum likelihood as

$$\min_X \sum_{i=1}^N \rho(e_i; X) \quad \text{or} \quad \sum_{i=1}^N \varphi(e_i; X) = 0 \quad \text{with} \quad \varphi(e_i; X) = \frac{d\rho(e_i; X)}{de_i}, \quad (44)$$

where  $\rho(e)$  is called the cost or penalty function [48] and must obey the properties:

$$\begin{aligned} & 1. \rho(e) \geq 0 \\ & 2. \rho(0) = 0 \\ & 3. \rho(e) = \rho(-e) \\ & 4. \rho(e_i) \geq \rho(e_j), |e_i| > |e_j|. \end{aligned} \quad (45)$$

Let

$$\rho_{MCC}(e_i) = \left(1 - \exp\left(-e_i^2/2\lambda^2\right)\right) / \sqrt{2\pi}\lambda. \quad (46)$$

It is trivial to check that it obeys the properties of Equation (45). Furthermore,

$$\begin{aligned} & \min \sum_{i=1}^N \rho_{MCC}(e_i) \\ &= \min \sum_{i=1}^N \left(1 - \exp\left(-e_i^2/2\lambda^2\right)\right) / \sqrt{2\pi}\lambda \\ &\Leftrightarrow \max \sum_{i=1}^N \exp\left(-e_i^2/2\lambda^2\right) / \sqrt{2\pi}\lambda \\ &= \max \sum_{i=1}^N G_\lambda(e_i). \end{aligned} \quad (47)$$

In other words, there is a connection with MCC and Equation (46) [50,51]:

$$\min_{\mathbf{X}} \sum_{i=1}^N \rho_{MCC}(e_i) \text{ is equivalent to } MCC = \max_{\mathbf{X}} \sum_{i=1}^N G_\lambda(e_i).$$

Considering the  $M$ -estimation problem, the weighting function  $\psi_{MCC}(e_i)$  is defined by

$$\psi_{MCC}(e_i) = \frac{\rho_{MCC}'(e_i)}{e_i}, \quad (48)$$

where  $\rho_{MCC}'(e_i) = \frac{d\rho_{MCC}(e_i)}{de_i}$ . Therefore,

$$\psi_{MCC}(e_i) = \exp\left(-e_i^2/2\lambda^2\right) / \sqrt{2\pi}\lambda^3. \quad (49)$$

Based on the penalty function and the weighting function above, we will analyze the robustness with influence function and compare it with Huber-filter to prove the superiority of MCCKF.

### 3.3.1. Influence Function of MCCKF

Influence function (IF) is a quantitative measure to evaluate the robustness of Kalman filter [52]. The conventional definition of IF is defined as:

**Definition 2.** For any  $l \in \mathbb{R}$ ,  $\mathbb{R}$  is the real set,  $\delta l$  is the probability distribution that is degraded in  $l$  and the mixture distribution is:

$$F = (1 - v)F_0 + v\delta l \quad (0 \leq v \leq 1), \quad (50)$$

where  $F_0$  is the main distribution, and  $v$  is the contamination ratio, and then if the limit

$$IF(l; F_0, T) = \lim_{v \rightarrow 0} \frac{T(F) - T(F_0)}{v} = \left. \frac{dT(F)}{dv} \right|_{v=0} \quad (51)$$

exists, where  $T(F)$  is one functional expression of the distribution function  $F$ , then  $IF(l; F_0, T)$  is called the influence function about  $T(F)$ .

The traditional definition of IF describes the relative influence of a contaminated measurement on functional  $T$ . The smaller the IF is, the more insensitive the reflection on the contaminated measurement of  $T$  is. In other words, for the fixed distribution  $F$ , the change of IF reflects the different influences on different measurements. Thus, it can effectively reflect the robustness of Kalman filter.

However, the traditional form of IF is sometimes complex to analyze the performance of Kalman filter. For analyzing the robustness of Kalman filter more clearly, Chang [48] derived a new form of the influence function. According to Equation (16), construct a following pseudo-measurement

$$\mathbf{Z}_k = \begin{bmatrix} \hat{\mathbf{X}}_{k|k-1} \\ \mathbf{Z}_k \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k \\ \mathbf{H}_k \mathbf{X}_k \end{bmatrix} + \begin{bmatrix} \delta \hat{\mathbf{X}}_{k|k-1} \\ \mathbf{V}_k \end{bmatrix} = \mathbf{H}_k \mathbf{X}_k + \mathbf{V}_k \quad (52)$$

with

$$E(\mathbf{V}_k) = 0, \quad E[(\mathbf{V}_k)(\mathbf{V}_k)^T] = \mathbf{R}_k = P_{\mathbf{V}_k} = \begin{bmatrix} P_{\hat{\mathbf{X}}_{k|k-1}, \hat{\mathbf{X}}_{k|k-1}} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_k \end{bmatrix}, \quad (53)$$

with  $\mathbf{Z}_k$  as the measurement and  $\mathbf{X}_k$  as the parameter to be estimated. Equation (52) is exactly a classic linear regression problem with the known statistics of the measurements, which can be solved using the weighted least square methods as

$$\hat{\mathbf{X}}_k = (\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k)^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{Z}_k, \quad (54)$$

$$P_{\hat{\mathbf{X}}_k \hat{\mathbf{X}}_k} = (\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k)^{-1}. \quad (55)$$

Define

$$\tilde{\mathbf{Z}}_k = \mathbf{R}_k^{-1/2} \mathbf{Z}_k, \quad (56)$$

$$\tilde{\mathbf{H}}_k = \mathbf{R}_k^{-1/2} \mathbf{H}_k, \quad (57)$$

$$\tilde{\mathbf{V}}_k = \mathbf{R}_k^{-1/2} \mathbf{V}_k, \quad (58)$$

where  $\mathbf{R}_k^{-1/2} = (\mathbf{R}_k^{1/2})^{-1}$  and  $\mathbf{R}_k^{1/2} (\mathbf{R}_k^{1/2})^T = \mathbf{R}_k$ . Then,

$$\tilde{\mathbf{Z}}_k = \tilde{\mathbf{H}}_k \mathbf{X}_k + \tilde{\mathbf{V}}_k, \quad (59)$$

$$E[\tilde{\mathbf{V}}_k] = 0, \quad P_{\tilde{\mathbf{V}}_k, \tilde{\mathbf{V}}_k} = \mathbf{I}. \quad (60)$$

For the sake of brevity, the subscript  $k$  is omitted and let  $m = n_x + n_y$  in the sequel, where  $n_x$ ,  $n_y$  denotes the dimension of the state vector and the measurement vector. Find the ML solution to Equation (59) through maximizing the following likelihood function

$$L(\hat{\mathbf{X}}, \mathbf{r}) = \prod_{j=1}^m F(\mathbf{r}_j), \quad (61)$$

where  $\mathbf{r} = \tilde{\mathbf{Z}}_k - \tilde{\mathbf{H}}_k \hat{\mathbf{X}}_k$ ,  $F(\cdot)$  denotes the probability distribution. Similar to Equation (41), Equation (61) is equal to the following form:

$$J(\hat{\mathbf{X}}, \mathbf{r}) = -\ln[L(\hat{\mathbf{X}}, \mathbf{r})] = \sum_{i=1}^m -\ln p(\mathbf{r}_i) = \sum_{i=1}^m \rho(\mathbf{r}_i), \quad (62)$$

where  $\rho(\cdot)$  is the cost or penalty function. Then, a necessary condition for the minimizer in Equation (62) is

$$\sum_{i=1}^m \rho'(r_i) \frac{\partial r_i}{\partial \hat{\mathbf{X}}} = \sum_{i=1}^m \phi(r_i) \frac{\partial r_i}{\partial \hat{\mathbf{X}}} = \sum_{i=1}^m \phi(r_i) \tilde{\mathbf{h}}_i^T = 0, \quad (63)$$

where

$$\phi(\cdot) = \rho'(\cdot) \quad (64)$$

and  $\tilde{\mathbf{h}}_i$  is the corresponding column vector of  $\tilde{\mathbf{H}}^T$  (the measurement matrix). By taking transpose, Equation (63) is equivalent to

$$\sum_{i=1}^m \phi(r_i) \tilde{\mathbf{h}}_i = 0. \quad (65)$$

Define an estimator as  $\mathcal{E}(F)$  explicating that it is a functional of the distribution  $F$ . Then, based on the description above, the new form of IF for the distribution Equation (43) is as follows:

$$IF(\tilde{\mathbf{h}}, F_0, \mathcal{E}) = \lim_{v \rightarrow 0} \frac{\mathcal{E}(F_0(1-v) + F_c v) - \mathcal{E}(F_0)}{v} = \left. \frac{\partial \mathcal{E}(F_v)}{\partial v} \right|_{v=0}, \quad (66)$$

where  $\tilde{\mathbf{h}}$  is the generalization of  $\tilde{\mathbf{h}}_j$  representing the corresponding column vector of  $\tilde{\mathbf{H}}^T$ ,  $v \downarrow 0$  stands for the limit from right. It evaluates the infinitesimal influence on the estimator  $\mathcal{E}$  of any single measurement whose residual is  $r$ , which comes from the neighborhood of the distribution  $F_0$  through the designing matrix  $\tilde{\mathbf{h}}^T$ . Let the method of estimating  $\hat{\mathbf{X}}$  for the  $m$ -sample data be denoted as  $\mathcal{E}_m$ . Then, according to the relation between  $\hat{\mathbf{X}}$  and  $\mathbf{r}$ , Chang [48] transformed the definition of IF in another form to obtain the IF of  $\mathcal{E}_m$ . It is derived in Ref. [48] in detail. The IF to assess the influence of one of the  $m$  elements of the pseudo-measurement vector in Equation (59) is defined as:

$$IF(r, \tilde{\mathbf{h}}, F_0, \mathcal{E}) = \frac{\phi(r_j)}{E_{F_0}[\phi'(r)]} \left( \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} \right)^{-1} \tilde{\mathbf{h}}_j, j = 1, 2, \dots, m, \quad (67)$$

where  $r_j$  is the  $j$ -th element of  $\mathbf{r}$ .

Three requirements about the IF are needed for an estimator to be infinitesimally robust, i.e., boundedness, continuity, and linearity at the center [48].

Next, we will prove that MCCKF satisfies the requirements above.

According to the Equation (46), the estimator of MCCKF with its cost function as

$$\phi_{MCC}(e) = \rho'_{MCC}(e) = \frac{e}{\sqrt{2\pi}\lambda^3} \exp\left(-\frac{e^2}{2\lambda^2}\right) \quad (68)$$

and

$$\phi'_{MCC}(e) = \frac{1}{\sqrt{2\pi}\lambda^3} \exp\left(-\frac{e^2}{2\lambda^2}\right) - \frac{e^2}{\sqrt{2\pi}\lambda^5} \exp\left(-\frac{e^2}{2\lambda^2}\right). \quad (69)$$

Then,

$$\begin{aligned} E_{F_0}[\phi'_{MCC}(e)] &= \int_{-\infty}^{+\infty} \phi'_{MCC}(e) dF_0 \\ &= \frac{(\lambda^2 + \sigma^2) - \sqrt{2\pi}}{\sqrt{2\pi}\lambda^2 (\lambda^2 + \sigma^2) \sqrt{\lambda^2 + \sigma^2}}, \end{aligned} \quad (70)$$

where  $\sigma$  is the standard deviation of distribution  $F_0$ . Therefore, we have the IF of MCC expressed as:

$$\begin{aligned} IF_{MCC} \left( r, \tilde{h}, F_0, \mathcal{E} \right) \\ = \frac{\phi_{MCC}(r_j)}{E_{F_0}[\phi'_{MCC}(r)]} \left( \tilde{\mathcal{H}}^T \tilde{\mathcal{H}} \right)^{-1} \tilde{h}_j, \end{aligned} \quad (71)$$

where  $E_{F_0}[\phi'_{MCC}(r)]$  is a constant (proved in Equation (70)) and  $(\tilde{\mathcal{H}}^T \tilde{\mathcal{H}})^{-1} \tilde{h}_j$  is a known constant matrix. It is trivial to check that the IF in Equation (71) satisfies continuity and linearity at the center. Next, we will prove the boundedness of Equation (71).

According to the statement above, the boundedness of IF depends on  $\phi_{MCC}(r_j)$ .

**Remark 1.**  $\phi_{MCC}(r)$  is a bounded function in the interval  $(-\infty, +\infty)$ .

**Proof.**

$$\begin{aligned} \phi_{MCC}(r) &= \frac{r}{\sqrt{2\pi}\lambda^3} \exp\left(-\frac{r^2}{2\lambda^2}\right) \\ &= \frac{r}{\sqrt{2\pi}\lambda^3} \bigg/ \exp\left(\frac{r^2}{2\lambda^2}\right). \end{aligned} \quad (72)$$

It is evident that  $\phi_{MCC}(r)$  is a continuous function, so it satisfies the boundedness in any closed intervals. Furthermore, it is an odd function.

1. Fix  $\varepsilon > 0$ , and  $\phi_{MCC}(r)$  is bounded in the interval  $[-\varepsilon, \varepsilon]$ .
2. In the interval  $[\varepsilon, +\infty)$ , according to the L'Hospital's rule [53], Equation (72) can be transformed as:

$$\begin{aligned} \lim_{r \rightarrow \infty} \phi_{MCC}(r) \\ = \lim_{r \rightarrow \infty} \frac{1}{\sqrt{2\pi}\lambda^3} \bigg/ \sqrt{2\pi}\lambda r \exp\left(\frac{r^2}{2\lambda^2}\right) \\ = 0. \end{aligned} \quad (73)$$

Therefore,  $\phi_{MCC}(r)$  is bounded in  $[\varepsilon, +\infty)$ . According to the characteristic of the odd function, it is also bounded in  $(-\infty, -\varepsilon]$ .  $\square$

Hence, the IF of MCCKF satisfies all three of the requirements of a robust estimator, i.e., boundedness, continuity, and linearity at the center.

HF is another filter whose robustness is also proved with IF in [48]. Next, we will compare the MCCKF with HF.

### 3.3.2. Comparison with Huber Filter

In fact, the Huber-based filter (HF) estimates by minimizing the following cost function [54]

$$J_{HF} = \min \left( \sum_{i=1}^N \rho_{HF}(e_i) \right), \quad (74)$$

where  $\rho_{HF}(\cdot)$  is the designed robust function that is bounded and convex. In other words, a Huber-based filter can resist the non-Gaussian noises because the robust function does not increase rapidly as the estimate residual grows. In addition, the  $M$ -estimation function with the expression shown in Equation (74) has several forms. Huber  $M$ -estimator [46] is a common one that can be shown as:

$$\rho_{\text{Huber}}(e_i) = \begin{cases} e_i^2/2, & |e_i| \leq \delta, \\ \delta(|e_i| - \delta/2), & |e_i| > \delta. \end{cases} \quad (75)$$

Its weighting function is

$$\psi_{\text{Huber}}(e_i) = \begin{cases} 1, & |e_i| \leq \delta, \\ \delta/|e_i|, & |e_i| > \delta, \end{cases} \quad (76)$$

where  $\delta$  is the tuning parameter. In a similar manner, Equation (46) can seem like a new  $\rho(\cdot)$  function in which the kernel size  $\lambda$  plays a role similar to  $\delta$ .

Suppose only one measurement is adopted. Figure 2 shows the  $\psi_{\text{MCC}}$  of the MCC in Equation (49) and  $\psi_{\text{Huber}}$  in Equation (76) with different errors, in which  $\lambda = 1$  and  $\delta = 1$ , respectively. The result is also presented in Ref. [46].

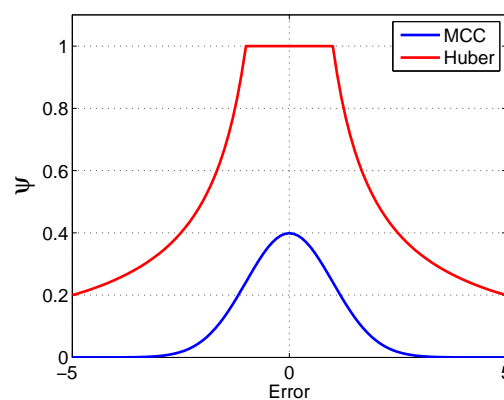


Figure 2.  $\psi$  with different estimation errors in different weighting functions.

The weighting function of MCC and Huber reduces as the error value increases. It means that both methods can reduce the weight of the measurements containing large outliers. However, MCC has a higher speed to zero compared to that of Huber. It means that Huber may employ more large outliers into the estimation process than that of MCC, although their impacts are reduced. In other words, MCC selects those measurements with smaller errors and discards those with a large outlier. MCC can completely remove the inappropriate measurements from the estimation processes and have a better performance. That is, Kalman filter under the MCC can improve the robustness.

From Equation (49), the kernel size  $\lambda$  plays an important role in the weighting function. The robustness is decided by the kernel size.

### 3.4. Kernel Size Selection

Due to the close relationship between  $M$ -estimation and correntropy, choosing an appropriate kernel size in the correntropy criterion becomes practical [54]. In the MCCKF algorithm, or the same input data, different kernel sizes lead to different filter results. If the kernel size is too large, the filter system will degrade into linear regression. If the kernel size is too small, all the data will look distinct and the system cannot infer on unseen samples falling between the training points [55]. Therefore, the kernel size can be adaptively adjusted as other parameters.

There are not many accessible values to the problem for us. However, the measurements, the dynamic system model and the noise parameters are accessible. It is a new thought to make use of this information in the adaption of the kernel size selection. In fact, Cinar et al. [56] has proposed that the best kernel sizes are in the range of the observation errors and half of the cost function has the

observation error as the argument and a similar kernel size will get a better result. Based on the idea, he heuristically chose the kernel size at each time instant to be the observation error as shown below:

$$\lambda_k = \left\| \mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k \right\|. \quad (77)$$

In this paper, inspired by Equation (77) and combined with the the weighted least square thought, we proposed that the kernel size selection method can be adjusted to

$$\lambda_{0_k} = \left\| \left( \mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k \right)^T \mathbf{R}_k^{-1} \left( \mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_k \right) \right\|. \quad (78)$$

In fact, Equations (77) and (78) reflect the Euclidean distance and the Mahalanobis distance, respectively. Equation (78), which is a novel kernel size selection method proposed in the article, can overcome the influence of the covariance due to the advantages of Mahalanobis distance. It is much more sensitive to the small change of data.

To show the different results of different kernel size, the weights are shown under the different sizes for  $\lambda$  in Equation (77) and for  $\lambda_0$  in Equation (78) as follows.

From Figure 3,  $\lambda_0$  has a higher speed to zero compared to  $\lambda$ .  $\lambda_0$  can employ less outliers than  $\lambda$ , although both impacts are reduced. Therefore,  $\lambda_0$  can select the measurements with smaller error and remove more inappropriate measurements from the estimation processes. Therefore,  $\lambda_0$  is more appropriate than  $\lambda$ .

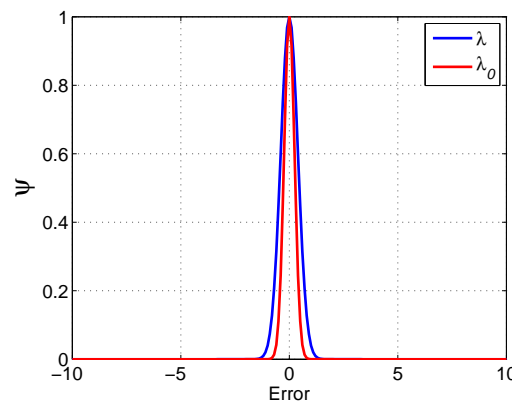


Figure 3.  $\psi$  with different kernel size selection methods (where  $\lambda_0$  is the novel kernel size selection method).

#### 4. Simulation

In this section, we mainly demonstrate the performances of the MCCKF in the presence of large outliers and mixture Gaussian noises. The estimation performances are compared based on the following indices:

$$\text{Residual}(k) = \mathbf{X}_k - \hat{\mathbf{X}}_{k|k}, \quad k = 1, \dots, N, \quad (79)$$

$$\text{RMSE} = \frac{1}{M} \sum_{m=1}^M \sqrt{\frac{1}{N} \sum_{i=1}^N \left( \mathbf{X}_k - \hat{\mathbf{X}}_{k|k} \right)^2}, \quad m = 1, \dots, M, \quad (80)$$

where  $N$  is the entire time steps in every Monte Carlo run and  $M$  represents the total Monte Carlo runs.

##### 4.1. Simulation Conditions

In this section, simulation experiments are conducted to validate the proposed algorithm for restricting the non-Gaussian noises. The  $\alpha$ -Jerk model results obtained by the Huber-based filter (HF),

ensemble Kalman filter (EnKF), unscented Kalman filter (UKF), Gaussian sum filter (GSF) and maximum correntropy criterion Kalman filter (MCKKF) are compared here.

The target tracking parameters are set as below: time interval is  $T = 1$  s; 500 independent Monte Carlo runs have been conducted, i.e.,  $M = 500$ ; the measurement noise covariance is  $R = \text{diag} \left( \begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \right)$ ; the standard deviation of input is  $\sigma_\varepsilon = 0.7$ ; the standard deviation of the target ‘jerk’ is  $\sigma_j = 0.2$ ; and the state vector in three dimensions is

$$\mathbf{X} = \begin{bmatrix} x & \dot{x} & \ddot{x} & \ddot{x} & y & \dot{y} & \ddot{y} & \ddot{y} & z & \dot{z} & \ddot{z} & \ddot{z} \end{bmatrix}^T,$$

which contains the position, the velocity, the acceleration and the ‘jerk’, respectively; the initial true state is

$$\begin{aligned} x(0) &= 30000, \dot{x}(0) = 300, \ddot{x}(0) = 1, \ddot{x}(0) = 0, \\ y(0) &= 60000, \dot{y}(0) = 400, \ddot{y}(0) = 1, \ddot{y}(0) = 0, \\ z(0) &= 90000, \dot{z}(0) = 500, \ddot{z}(0) = 1, \ddot{z}(0) = 0. \end{aligned}$$

and the initial covariance matrix of the states are chosen as

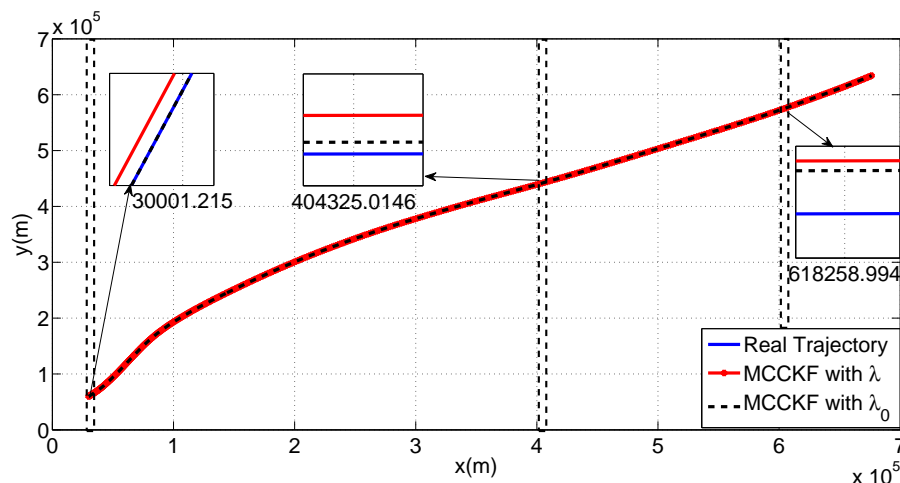
$$\mathbf{P}_{0|0} = \text{diag} \left( \begin{bmatrix} 10000 & 1000 & 100 & 10 & 10000 & 1000 & 100 & 10 & 10000 & 1000 & 100 & 10 \end{bmatrix} \right).$$

The initial values in three dimensions are  $\alpha_x = \alpha_y = \alpha_z = 0.5$ .

#### 4.2. The Kernel Size Adaptive Method

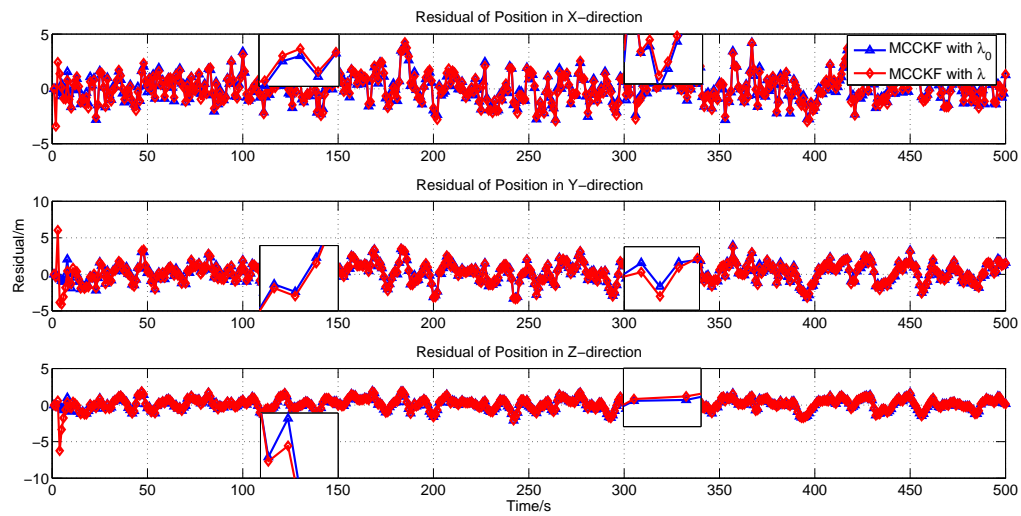
To choose a better kernel size adaptive method, a simulation is made in this section with Equations (77) and (78), respectively. The simulation conditions are the same as Section 4.1. Figure 4 shows the trajectories in a two-dimensional plane under two kernel size selection methods.

As can be seen from Figures 4 and 5, MCKKF with the new adaptive method ensures a higher accuracy than the other in most time.



**Figure 4.** Target trajectories of two kernel size adaptive methods on  $\alpha$ -jerk model where  $\lambda_0$  is the novel kernel size adaptive method.





**Figure 5.** Residual of estimated position with two kernel size adaptive methods where  $\lambda_0$  is the novel kernel size adaptive method.

Tables 1–3 show that the newly proposed kernel size selection method in this paper ( $\lambda_0$ ) indeed has a higher accuracy on the whole.

**Table 1.** RMSE of every state variable in the  $x$ -direction of two different kernel size selection methods.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
$\lambda_0$	<b>1.2331</b>	<b>1.3251</b>	<b>0.7766</b>	<b>0.3512</b>	<b>0.2637</b>
$\lambda$	1.2821	1.3590	0.7961	0.3531	0.2764

**Table 2.** RMSE of every state variable in the  $y$ -direction of two different kernel size selection methods.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
$\lambda_0$	<b>1.2325</b>	<b>1.2203</b>	<b>0.6863</b>	<b>0.3248</b>	<b>0.3232</b>
$\lambda$	1.2538	1.2451	0.6905	0.32	0.3232

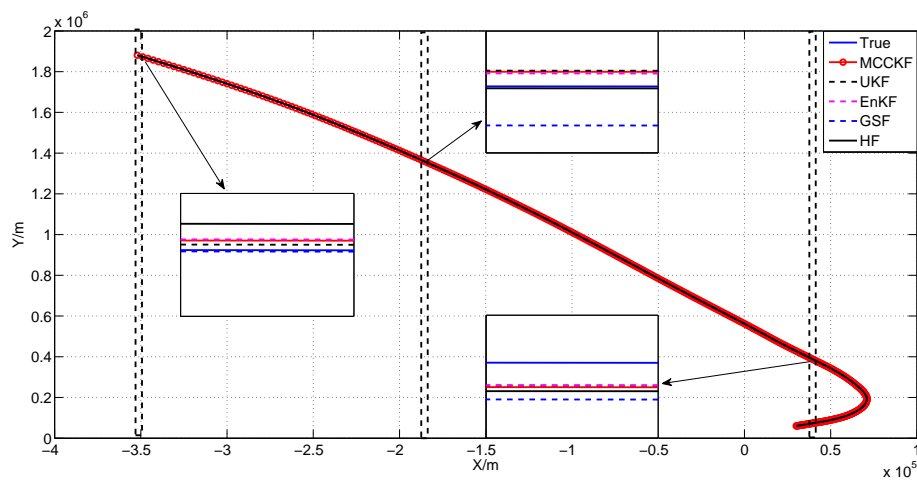
**Table 3.** RMSE of every state variable in the  $z$ -direction of two different kernel size selection methods.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
$\lambda_0$	<b>1.3477</b>	<b>1.3768</b>	<b>0.8037</b>	<b>0.3581</b>	<b>0.7747</b>
$\lambda$	1.3747	1.3968	0.8256	0.3682	0.7936

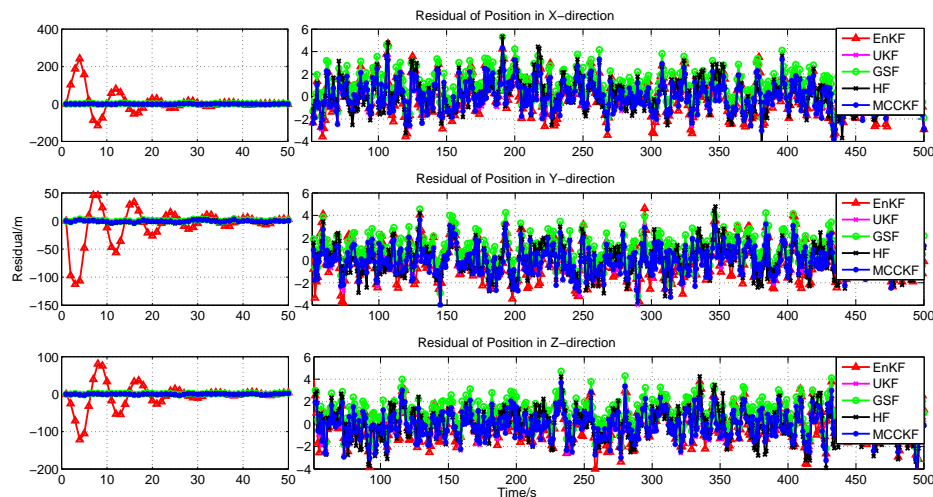
#### 4.3. The Presence of Gaussian Noise

In this section, we will apply the MCCKF on  $\alpha$ -jerk model in the presence of Gaussian noises to present the performance of MCCKF. The simulation parameters and the initial state are the same as Section 4.1.

To explain the advantages of MCCKF, we use EnKF, UKF, GSF and HF algorithms to track target with  $\alpha$ -jerk model compared with MCCKF. Figure 6 demonstrates the real target trajectory and the estimated target trajectories with EnKF, GSF, UKF, HF and MCCKF in a two-dimensional plane. Figure 7 demonstrates the residual defined in Equation (79) with EnKF, UKF, GSF, HF and MCCKF. Tables 4–6 summarize the corresponding RMSE of every state variable with five algorithms in three directions.



**Figure 6.** Target trajectories of different filter algorithms on the  $\alpha$ -jerk model in the presence of Gaussian noises.



**Figure 7.** Residual of estimated position with different filter algorithms in the presence of Gaussian noises.

As can be seen from Figures 6 and 7, MCKKF does not have the best performance most of the time. Furthermore, it can be seen that the performance of MCKKF may be worse than UKF in general combined with Tables 4–6.

**Table 4.** RMSE of every state variable in the  $x$ -direction of EnKF, UKF, GSF, HF and MCKKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	7.0350	4.7926	2.3029	0.5543	1.7291
UKF	1.2892	1.3825	0.8940	0.4424	1.6125
GSF	1.5990	1.2220	0.8072	0.4202	1.6125
HF	1.2843	1.2220	0.8072	0.4202	1.6125
MCKKF	1.2901	1.2252	0.8076	0.4201	1.6125

**Table 5.** RMSE of every state variable in the  $y$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	3.2812	3.9866	2.1125	0.5019	3.2963
UKF	1.2856	1.5343	0.9418	0.4299	1.1582
GSF	1.5749	1.3181	0.8666	0.4245	1.1582
HF	1.2695	1.3181	0.8666	0.4245	1.1582
<b>MCCKF</b>	<b>1.2695</b>	<b>1.3195</b>	<b>0.8685</b>	<b>0.4245</b>	<b>1.1582</b>

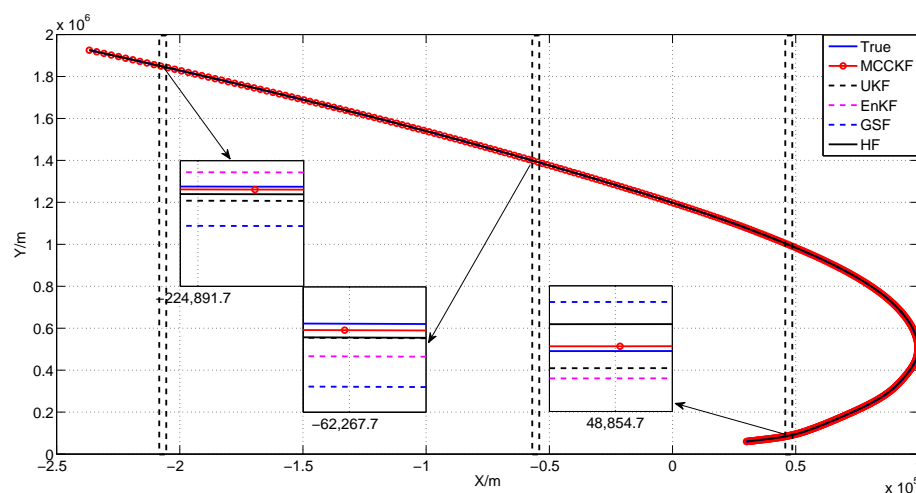
**Table 6.** RMSE of every state variable in the  $z$ -direction of EnKF, UKF, GSF and MCCKF algorithms.

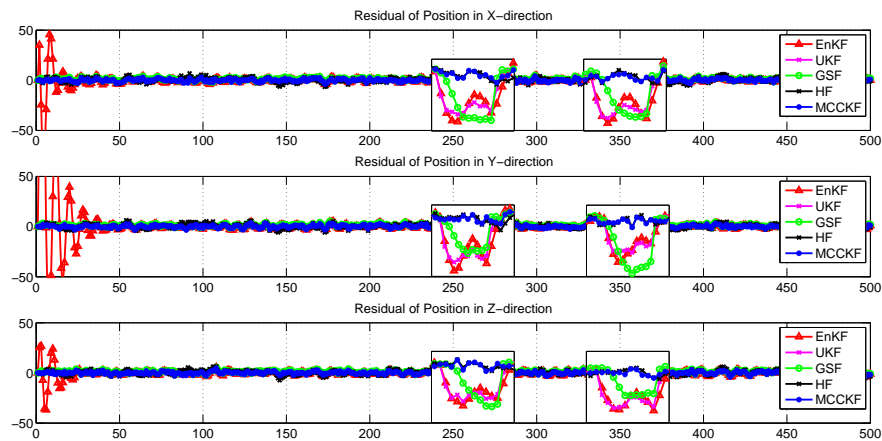
	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	4.5702	3.5353	1.9189	0.5649	1.9978
UKF	1.2178	1.4724	0.9319	0.4700	1.1227
GSF	1.5931	1.2563	0.8887	0.4738	1.1227
HF	1.2071	1.2562	0.8887	0.4738	1.1227
<b>MCCKF</b>	<b>1.2125</b>	<b>1.2617</b>	<b>0.8907</b>	<b>0.4741</b>	<b>1.1227</b>

#### 4.4. The Presence of Large Outliers

In this section, we will apply the MCCKF on  $\alpha$ -Jerk model in the presence of large outliers to verify the robustness of MCCKF. The simulation parameters and the initial state are the same as Section 4.1. The constant outliers are added, respectively, from 251 s to 260 s and from 351 s to 360 s. The magnitude of the outliers is 10.

To explain the advantages of MCCKF, we use the EnKF, UKF, GSF, and HF algorithms to track targets with  $\alpha$ -jerk models compared with MCCKF. Figure 8 demonstrates the real target trajectory and the estimated target trajectories with EnKF, GSF, UKF, HF and MCCKF in a two-dimensional plane. Figure 9 demonstrates the residual defined in Equation (79) with EnKF, UKF, GSF, HF and MCCKF. Tables 7–9 summarize the corresponding RMSE of every state variable with five algorithms in three directions.

**Figure 8.** Target trajectories of different filter algorithms on  $\alpha$ -jerk model in the presence of large outliers.



**Figure 9.** Residual of estimated position with different filter algorithms in the presence of large outliers.

As can be seen from Figure 8, MCCKF has a higher accuracy in most time. From Figure 9, it can be evidently seen that EnKF, UKF and GSF have divergence during the time of outliers. HF and MCCKF have a stable behavior in the whole process. Tables 7–9 show that MCCKF has a better performance than others, including HF.

**Table 7.** RMSE of every state variable in  $x$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	6.9390	6.4846	3.7753	0.6530	1.3750
UKF	3.3104	1.7170	0.9083	0.4238	0.9707
GSF	3.0822	1.5156	0.8306	0.4258	0.9707
HF	2.3903	1.4708	1.7734	1.3180	1.0783
<b>MCCKF</b>	<b>1.4252</b>	<b>1.3336</b>	<b>0.8153</b>	<b>0.4245</b>	<b>0.9707</b>

**Table 8.** RMSE of every state variable in  $y$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	3.7977	2.8730	1.7031	0.5630	5.5318
UKF	3.3085	1.6968	0.9586	0.4464	0.3219
GSF	3.5637	1.6479	0.8553	0.4259	0.3219
HF	2.2420	3.2366	1.6525	1.3194	0.8478
<b>MCCKF</b>	<b>1.3119</b>	<b>1.2571</b>	<b>0.8020</b>	<b>0.4244</b>	<b>0.3219</b>

**Table 9.** RMSE of every state variable in  $z$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	11.6641	9.4672	4.7836	0.8071	2.8136
UKF	3.1840	1.6303	0.9613	0.4324	0.6726
GSF	2.8279	1.3632	0.8259	0.4192	0.6726
HF	2.2469	3.2751	1.6901	1.3161	0.9380
<b>MCCKF</b>	<b>1.3696</b>	<b>1.2799</b>	<b>0.8209</b>	<b>0.4188</b>	<b>0.6726</b>

#### 4.5. The Presence of Gaussian Mixture Noises

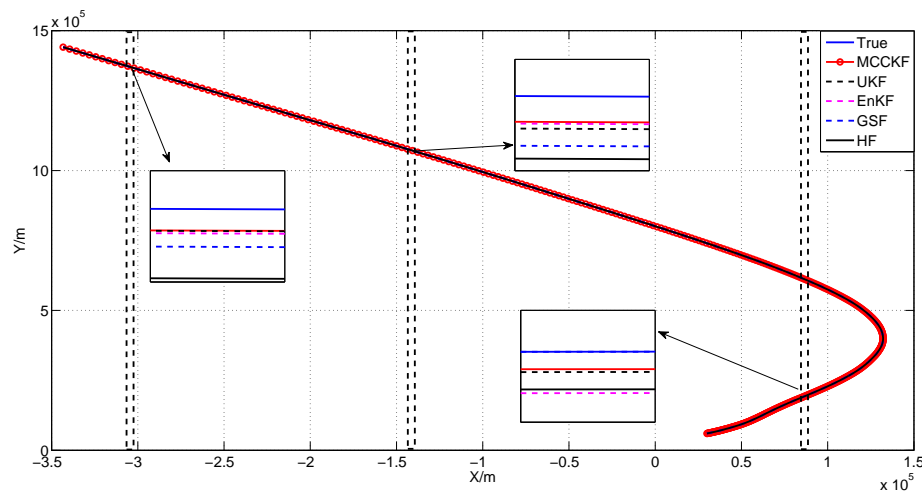
In this section, we will apply the MCCKF on the  $\alpha$ -Jerk model in the presence of Gaussian mixture noises to verify the advantage of MCCKF. The simulation parameters and the initial state are also the

same as Section 4.1. In the measurement process, we add the Gaussian mixture noise. The Gaussian mixture noise model is as follows:

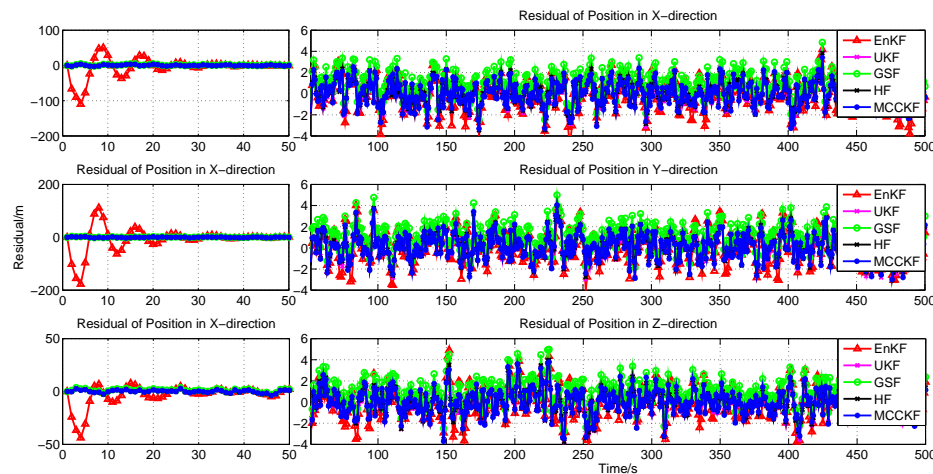
$$V_k = (1 - v) N(\mu_1, \sigma_1^2) + v N(\mu_2, \sigma_2^2), \quad (81)$$

where  $\mu_1 = [0 \ 0 \ 0]$ ,  $\mu_2 = [0 \ 0 \ 0]$ ,  $\sigma_1^2 = R$ ,  $\sigma_2^2 = \text{diag}([10 \ 10 \ 10])$  and  $v = 0.01$ .

The accuracy of the MCCKF is compared with the EnKF, UKF, GSF, and HF. Figure 10 demonstrates the real target trajectory and the estimated target trajectories with EnKF, UKF, GSF, HF and MCCKF. Figure 11 demonstrates the residual defined in Equation (79) with EnKF, GSF, UKF, HF and MCCKF in a two-dimensional plane. Tables 8–10 summarize the corresponding RMSE of every state variable with five algorithms in three directions.



**Figure 10.** Target trajectories of different filter algorithms on the  $\alpha$ -jerk model in the presence of Gaussian mixture noises.



**Figure 11.** Residual of estimated position with different filter algorithms in the presence of Gaussian mixture noises.

Similar to the presence of large outliers, MCCKF has a higher accuracy in most time compared to other methods from Figures 10 and 11. Furthermore, it can be seen that it has a better performance than others from Tables 10–12.

**Table 10.** RMSE of every state variable in the  $x$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	6.4009	6.4574	3.3219	0.5756	2.0893
UKF	1.3171	1.5839	1.0426	0.4578	0.9669
GSF	1.6669	1.4316	0.9840	0.4498	0.9669
HF	1.5411	1.4907	0.9834	0.4703	0.9669
<b>MCCKF</b>	<b>1.3080</b>	<b>1.4194</b>	<b>0.9714</b>	<b>0.4497</b>	<b>0.9669</b>

**Table 11.** RMSE of every state variable in the  $y$ -direction of EnKF, UKF, GSF, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	5.6661	6.9973	3.8323	0.6798	1.6025
UKF	1.2481	1.4203	0.9216	0.4233	1.5972
GSF	1.6537	1.2715	0.8655	0.412	1.5972
HF	1.2348	1.2715	0.8655	0.4122	1.5972
<b>MCCKF</b>	<b>1.0214</b>	<b>1.2607</b>	<b>0.8226</b>	<b>0.4117</b>	<b>1.5972</b>

**Table 12.** RMSE of every state variable in the  $z$ -direction of EnKF, UKF, GS, HF and MCCKF algorithms.

	Position (m)	Velocity (m/s)	Acceleration (m/s <sup>2</sup> )	Jerk (m/s <sup>3</sup> )	$\alpha$
EnKF	2.4590	2.6990	1.5505	0.5314	3.0559
UKF	1.2806	1.5766	1.0119	0.4717	0.4726
GSF	1.5664	1.3625	0.9533	0.4678	0.4726
HF	1.2877	1.3625	0.9529	0.4678	0.4726
<b>MCCKF</b>	<b>1.2548</b>	<b>1.3602</b>	<b>0.9134</b>	<b>0.4678</b>	<b>0.4726</b>

## 5. Conclusions

$\alpha$ -jerk model, a modified jerk model, can estimate the parameter  $\alpha$  in real time during the process of target tracking. In the process of tracking targets, the measurement noise is a research focus. Large outliers and Gaussian mixture noises are the most popular non-Gaussian noises in the measurement process. Maximum Correntropy Criterion Kalman Filter is deduced with the weighted least square algorithm, which is proved to be rational based on the correntropy theory. Its robustness in the state estimation is analyzed with influence function for the first time. It proves that the filter algorithm can overcome the influence of non-Gaussian noises. Furthermore, a novel kernel size selection is proposed, which can improve the robustness of the filter algorithm. Simulated experimental results demonstrate that it provides effective performance to control the non-Gaussian noises of  $\alpha$ -jerk model.

**Acknowledgments:** This work was supported in part by the National Defense Technology Foundation of China under Grant No. 3101065, the National Natural Science Foundation of China (NSFC) under Grant Nos. 61773021, 61703408, and 61573367 and the Student Research Innovation Scholarship of Hunan Province under Grant No. CX2014B010.

**Author Contributions:** Bowen Hou proposed the main idea, wrote the draft manuscript and presented the experiments; Zhangming He and Dong Li conceived of the experiments; Xuanying Zhou and Haiyin Zhou analyzed the data and drew the figures and tables; Jiongqi Wang carried out the simulations and revised the manuscript.

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

## References

- Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *39*, 1333–1364. [[CrossRef](#)]
- Yang, X.; Xing, K.; Feng, X. Maneuvering target tracking in dense clutter based on particle filtering. *Chin. J. Aeronaut.* **2011**, *24*, 171–180. [[CrossRef](#)]

3. Singer, R.A. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.* **1970**, AES-6, 473–483. [[CrossRef](#)]
4. Bar-Shalom, Y.; Birmiwal, K. Variable dimension filter for maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **1982**, AES-18, 621–629. [[CrossRef](#)]
5. Kumar, K.S.P.; Zhou, H.; Kumar, K.S.P.; Zhou, H. A “current” statistical model and adaptive algorithm for estimating maneuvering targets. *J. Guidance Control Dyn.* **1984**, 7, 596–602. [[CrossRef](#)]
6. Mehrotra, K.; Mahapatra, P.R. A jerk model for tracking highly maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, 33, 1094–1105. [[CrossRef](#)]
7. Luo, X.B. A  $\alpha$ -jerk model for tracking maneuvering targets. *Signal Process.* **2007**, 4, 481–484.
8. Li, X.R.; Jilkov, V.P. A survey of maneuvering target tracking: Approximation techniques for nonlinear filtering. *Proc. SPIE* **2004**, 5428, doi:10.1117/12.553357. [[CrossRef](#)]
9. Roth, M.; Hendeby, G.; Fritsche, C.; Gustafsson, F. The ensemble kalman filter: A signal processing perspective. *Eurasip J. Adv. Signal Process.* **2017**, 2017, 56. [[CrossRef](#)]
10. Ammann, N.; Andert, F. Visual navigation for autonomous, precise and safe landing on celestial bodies using unscented kalman filtering. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017. [[CrossRef](#)]
11. Zhang, C.; Hwang, I. Gaussian sum-based maneuvering target tracking using unmanned aerial vehicle. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Grapevine, TX, USA, 9–13 January 2017; pp. 1–12. [[CrossRef](#)]
12. Julier, S.J.; Uhlmann, J.K.; Durrant-Whyte, H.F. A new approach for filtering nonlinear systems. In Proceedings of the American Control Conference, Seattle, WA, USA, 21–23 June 1995; Volume 3, pp. 1628–1632. [[CrossRef](#)]
13. Uhlmann, J.K.; Julier, S.J. A new extension of the kalman filter to nonlinear systems. *Proc. SPIE* **1997**, 3068, 182–193.
14. Maybeck, P.S. *Stochastic Models, Estimation and Control Volume 3*; Academic Press: New York, NY, USA, 1979.
15. Stein, D.W.J. Detection of random signals in gaussian mixture noise. In Proceedings of the 1994 28th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 31 October–2 November 1994; Volume 2, pp. 791–795. [[CrossRef](#)]
16. Wu, W.R. Target tracking with glint noise. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, 29, 174–185. [[CrossRef](#)]
17. Plataniotis, K.N.; Androutsos, D.; Venetsanopoulos, A.N. Nonlinear filtering of non-gaussian noise. *J. Intell. Robot. Syst.* **1997**, 19, 207–231. [[CrossRef](#)]
18. Sorenson, H.W.; Alspach, D.L. Recursive bayesian estimation using gaussian sums. *Automatica* **1971**, 7, 465–479. [[CrossRef](#)]
19. Harvey, A.; Luati, A. Filtering with heavy tails. *J. Am. Stat. Assoc.* **2014**, 109, 1112–1122. [[CrossRef](#)]
20. Izanloo, R.; Fakoorian, S.A.; Yazdi, H.S.; Dan, S. Kalman filtering based on the maximum correntropy criterion in the presence of non-gaussian noise. In Proceedings of the Information Science and Systems, Princeton, NJ, USA, 16–18 March 2016; pp. 500–505. [[CrossRef](#)]
21. Wan, E.A.; van der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 4 October 2000; pp. 153–158. [[CrossRef](#)]
22. Curn, J.; Marinescu, D.; Lacey, G.; Cahill, V. Estimation with non-white gaussian observation noise using a generalised ensemble kalman filter. In Proceedings of the IEEE International Symposium on Robotic and Sensors Environments, Magdeburg, Germany, 16–18 November 2012; pp. 85–90. [[CrossRef](#)]
23. Kotecha, J.H.; Djuric, P.M. Gaussian sum particle filtering. *IEEE Trans. Signal Process.* **2003**, 51, 2602–2612. [[CrossRef](#)]
24. Liu, Y.; Dong, K.; Wang, H.; Liu, J.; He, Y.; Pan, L. Adaptive gaussian sum squared-root cubature kalman filter with split-merge scheme for state estimation. *Chin. J. Aeronaut.* **2014**, 27, 1242–1250. [[CrossRef](#)]
25. Yin, J.; Zhang, J.; Zhuang, Z. Gaussian sum phd filtering algorithm for nonlinear non-gaussian models. *Chin. J. Aeronaut.* **2008**, 21, 341–351. [[CrossRef](#)]
26. Rousseeuw, P.J.; Leroy, A.M. Robust Regression and Outlier Detection. *J. R. Stat. Soc.* **1989**, 152, 133–134. [[CrossRef](#)]



27. Wang, R.; Xiong, Z.; Liu, J.Y.; Li, R.; Peng, H. SINS/GPS/CNS information fusion system based on improved huber filter with classified adaptive factors for high-speed UAVs. In Proceedings of the 2012 IEEE/ION Position Location and Navigation Symposium (PLANS), Myrtle Beach, SC, USA, 23–26 April 2012; pp. 441–446. [\[CrossRef\]](#)
28. Huang, J.; Yan, B.; Hu, S. Centralized fusion of unscented kalman filter based on huber robust method for nonlinear moving target tracking. *Math. Probl. Eng.* **2015**, *2015*, 291913. [\[CrossRef\]](#)
29. Chang, L.; Li, K.; Hu, B. Huber's m-estimation-based process uncertainty robust filter for integrated ins/gps. *IEEE Sens. J.* **2015**, *15*, 3367–3374. [\[CrossRef\]](#)
30. Chien-Hao, T.; Lin, S.F.; Dah-Jing, J. Robust huber-based cubature kalman filter for gps navigation processing. *J. Navigat.* **2016**, *70*, 527–546. [\[CrossRef\]](#)
31. Chang, G. Robust kalman filtering based on mahalanobis distance as outlier judging criterion. *J. Geodesy* **2014**, *88*, 391–401. [\[CrossRef\]](#)
32. Rodger, J.A. Toward reducing failure risk in an integrated vehicle health maintenance system: A fuzzy multi-sensor data fusion kalman filter approach for ivhms. *Expert Syst. Appl.* **2012**, *39*, 9821–9836.
33. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum correntropy kalman filter. *Automatica* **2017**, *76*, 70–77. [\[CrossRef\]](#)
34. Liu, X.; Qu, H.; Zhao, J.; Yue, P.; Wang, M. Maximum correntropy unscented kalman filter for spacecraft relative state estimation. *Sensors* **2016**, *16*, 1530. [\[CrossRef\]](#)
35. Liu, X.; Chen, B.; Xu, B.; Wu, Z.; Honeine, P. Maximum correntropy unscented filter. *Int. J. Syst. Sci.* **2017**, *48*, 1607–1615. [\[CrossRef\]](#)
36. Xi, L.; Hua, Q.; Zhao, J.; Chen, B. Extended Kalman filter under maximum correntropy criterion. In Proceedings of the International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016; pp. 1733–1737. [\[CrossRef\]](#)
37. Wang, J.J.-Y.; Wang, Y.; Jing, B.-Y.; Gao, X. Regularized maximum correntropy machine. *Neurocomputing* **2015**, *160*, 85–92. [\[CrossRef\]](#)
38. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part V. multiple-model methods. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1255–1321. [\[CrossRef\]](#)
39. Lewis, F.L. Applied optimal control. *IEEE Trans. Autom. Control* **2003**, *17*, 186–188.
40. Jiang, C.; Zhang, S.B.; Zhang, Q.Z. A new adaptive h-infinity filtering algorithm for the GPS/INS integrated navigation. *Sensors* **2016**, *16*, 21–27. [\[CrossRef\]](#)
41. Príncipe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer Publishing Company, Incorporated: Berlin, Germany, 2010.
42. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Príncipe, J.C. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884.
43. Chen, B.; Wang, J.; Zhao, H.; Zheng, N.; Príncipe, J.C. Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Process. Lett.* **2015**, *22*, 1723–1727.
44. Chen, B.; Príncipe, J.C. Maximum correntropy estimation is a smoothed map estimation. *IEEE Signal Process. Lett.* **2012**, *19*, 491–494.
45. Liu, W.; Pokharel, P.P.; Príncipe, J.C. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [\[CrossRef\]](#)
46. He, R.; Hu, B.; Yuan, X.; Wang, L. *Robust Recognition via Information Theoretic Learning*; Springer International Publishing: Berlin, Germany, 2014.
47. Huber, P.J. Robust estimation of a location parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [\[CrossRef\]](#)
48. Chang, G.; 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. [\[CrossRef\]](#)
49. Chang, G.; Xu, T.; Wang, Q. M-estimator for the 3d symmetric helmert coordinate transformation. *J. Geodesy* **2017**, *2017*, 1–12. [\[CrossRef\]](#)
50. He, Y.; Wang, F.; Yang, J.; Rong, H.; Chen, B. Kernel adaptive filtering under generalized maximum correntropy criterion. In Proceedings of the International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016; pp. 1738–1745. [\[CrossRef\]](#)
51. Wang, Y.; Zheng, W.; Sun, S.; Li, L. Robust information filter based on maximum correntropy criterion. *J. Guidance Control Dyn.* **2016**, *39*, 1–6. [\[CrossRef\]](#)
52. Huber, P.J. The 1972 wald lecture robust statistics: A review. *Ann. Math. Stat.* **1972**, *43*, 1041–1067. [\[CrossRef\]](#)



53. Taylor, A.E. L'hospital's rule. *Am. Math. Mon.* **1952**, *59*, 20–24. [[CrossRef](#)]
54. He, R.; Zheng, W.S.; Hu, B.G. Maximum correntropy criterion for robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1561–1576. [[CrossRef](#)]
55. Liu, W.; Príncipe, J.C.; Haykin, S.S. *Kernel Adaptive Filtering: A Comprehensive Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
56. Cinar, G.T.; Príncipe, J.C. Hidden state estimation using the correntropy filter with fixed point update and adaptive kernel size. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012. [[CrossRef](#)]



© 2017 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/>).