Maximum Correntropy Unscented Filter

Xi Liu, Badong Chen, Bin Xu, Zongze Wu, Paul Honeine

To appear in the *International Journal of Systems Science*
Vol. 00, No. 00, Month 20XX, 1–18

Maximum Correntropy Unscented Filter

Xi Liu[a], Badong Chen[a*], Bin Xu[b], Zongze Wu[c], and Paul Honeine[d]

[a]*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China;* [b]*School of Automation, Northwestern Polytechnical University, Xi'an, China;* [c]*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China;* [d]*the Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, Rouen, France.*

(*v5.0 released February 2015*)

The unscented transformation (UT) is an efficient method to solve the state estimation problem for a non-linear dynamic system, utilizing a derivative-free higher-order approximation by approximating a Gaussian distribution rather than approximating a non-linear function. Applying the UT to a Kalman filter type estimator leads to the well-known unscented Kalman filter (UKF). Although the UKF works very well in Gaussian noises, its performance may deteriorate significantly when the noises are non-Gaussian, especially when the system is disturbed by some heavy-tailed impulsive noises. To improve the robustness of the UKF against impulsive noises, a new filter for nonlinear systems is proposed in this work, namely the maximum correntropy unscented filter (MCUF). In MCUF, the UT is applied to obtain the prior estimates of the state and covariance matrix, and a robust statistical linearization regression based on the maximum correntropy criterion (MCC) is then used to obtain the posterior estimates of the state and covariance matrix. The satisfying performance of the new algorithm is confirmed by two illustrative examples.

Keywords: Unscented Kalman Filter (UKF); Unscented Transformation (UT); Maximum Correntropy Criterion (MCC).

1. Introduction

Estimation problem plays a key role in many fields, including communication, navigation, signal processing, optimal control and so on (Dash, Hasan, & Panigrahi, 2010; D. Li, Liu, Qiao, & Xiong, 2010; L. Li & Xia, 2013; Partovibakhsh & Liu, 2014). The Kalman filter (KF) assists in obtaining accurate state estimation for a linear dynamic system, which provides an optimal recursive solution under minimum mean square error (MMSE) criterion (Bryson & Ho, 1975; Kalman, 1960; Nahi, 1969). Nevertheless, most practical systems are inherently nonlinear, and it is not easy to implement an optimal filter for nonlinear systems. To solve the nonlinear filtering problem, so far many sub-optimal nonlinear extensions of the KF have been developed by using some approximations, among which the extended Kalman filter (EKF) (Anderson & Moore, 1979) and unscented Kalman filter (UKF) (Julier, Uhlmann, & Durrant-Whyte, 2000) are two widely used ones. As a popular nonlinear extension of KF, the EKF approximates the nonlinear system by its first order linearization and uses the original KF on this approximation. However, the crude approximation may lead to divergence of the filter when the function is highly non-linear. Moreover, the cumbersome derivation of the Jacobian matrices often leads to the implementation difficulties. The UKF is an alternative to the EKF, which approximates the probability distribution of the state by a set of deterministically chosen sigma points and propagates the distribution though

*Corresponding author. Email: chenbd@mail.xjtu.edu.cn

the non-linear equations. The UKF does not need to calculate the Jacobian matrices and can obtain a better performance than the EKF. However, the UKF may performs poorly when the system is disturbed by some heavy-tailed non-Gaussian noises, which occur frequently in many real-world applications of engineering. The main reason for this is that the UKF is based on the MMSE criterion and thus exhibits sensitivity to heavy-tailed noises (Schick & Mitter, 1994). Some methods have been proposed to cope with this problem in the literatures. In particular, the Huber's generalized maximum likelihood methodology is an important one (El-Hawary & Jing, 1995; Huber, 1964; Karlgaard & Schaub, 2007; X. Wang, Cui, & Guo, 2010), which is a combined minimum ℓ_1 and ℓ_2 norm estimation method. Furthermore, the statistical linear method instead of the first order linearization was introduced in (Karlgaard & Schaub, 2007).

Besides Huber's robust statistics, information theoretic quantities (e.g. entropy, mutual information, divergence, etc.) can also be used as a robust cost for estimation problems (B. Chen, Zhu, Hu, & Principe, 2013; Principe, 2010). As a localized similarity measure in information theoretic learning (ITL), the correntropy has recently been successfully applied in robust machine learning and non-Gaussian signal processing (Bessa, Miranda, & Gama, 2009; B. Chen & Principe, 2012; B. Chen, Xing, Liang, Zheng, & Principe, 2014; B. Chen, Xing, Zhao, Zheng, & Principe, 2016; X. Chen, Yang, Liang, & Ye, 2012; He, Hu, Yuan, & Wang, 2014; He, Hu, Zheng, & Kong, 2011; He, Zheng, & Hu, 2011; Liu, Pokharel, & Principe, 2007; Shi & Lin, 2014; Singh & Principe, 2009; Y. Wang, Pan, Xiang, & Zhu, 2015; Xu & Principe, 2008; Zhao, Chen, & Principe, 2011). The adaptive filtering algorithms under the maximum correntropy criterion (MCC) can achieve excellent performance in heavy-tailed non-Gaussian noises (B. Chen et al., 2014, 2016; Principe, 2010; Shi & Lin, 2014; Singh & Principe, 2009; Y. Wang et al., 2015; Zhao et al., 2011). In particular, in a recent work (B. Chen, Liu, Zhao, & Principe, 2015), a linear Kalman type filter has been developed under MCC, which can outperform the original KF significantly especially when the disturbance noises are impulsive.

The goal of this paper is to develop an unscented non-linear Kalman type filter based on the MCC, called the maximum correntropy unscented filter (MCUF). In the MCUF, the unscented transformation (UT) is applied to get a prior estimation of the state and covariance matrix, and a statistical linearization regression model based on MCC is used to obtain the posterior state and covariance. The new filter adopts the UT and statistical linear approximation instead of the first order approximation as in EKF to approximate the nonlinearity, and uses the MCC instead of the MMSE to cope with the non-Gaussianity, and hence can achieve desirable performance in highly nonlinear and non-Gaussian conditions. Moreover, the proposed MCUF is suitable for online implementation on account of the retained recursive structure.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the MCC. In Section 3, we derive the MCUF algorithm. In Section 4, we present two illustrative examples and show the desirable performance of the proposed algorithm. Finally, Section 5 concludes this paper.

2. Maximum correntropy criterion

Correntropy is a generalized similarity measure between two random variables. Given two random variables $X, Y \in \mathbb{R}$ with joint distribution function $\mathrm{F}_{XY}(x, y)$, the correntropy is defined by

$$V(X,Y) = \mathrm{E}\left[\kappa(X,Y)\right] = \int \kappa(x,y) d\mathrm{F}_{XY}(x,y) \tag{1}$$

where E denotes the expectation operator, and $\kappa(\cdot,\cdot)$ is a shift-invariant Mercer kernel. In this work, without mentioned otherwise, the used kernel function is the Gaussian kernel:

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

where $e = x - y$, and $\sigma > 0$ stands for the kernel bandwidth.

In many practical situations, we have only a set of finite data and the joint distribution F_{XY} is unknown. In these cases, one can estimate the correntropy using a sample mean estimator:

$$\widehat{V}(X,Y) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{G}_\sigma\left(e(i)\right) \tag{3}$$

where $e(i) = x(i) - y(i)$, with $\{x(i), y(i)\}_{i=1}^{N}$ being N samples drawn from F_{XY}.

Using the Taylor series expansion for the Gaussian kernel yields

$$V(X,Y) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n \sigma^{2n} n!} \mathrm{E}\left[(X-Y)^{2n}\right] \tag{4}$$

Thus, the correntropy is a weighted sum of all even order moments of the error variable $X - Y$. The kernel bandwidth appears as a parameter to weight the second order and higher order moments. With a very large kernel bandwidth (compared to the dynamic range of the data), the correntropy will be dominated by the second order moment.

Suppose our goal is to learn a parameter vector W of an adaptive model, and let $x(i)$ and $y(i)$ denote, respectively, the model output and the desired response. The MCC based learning can be formulated as solving the following optimization problem:

$$\widehat{W} = \arg\max_{W\in\Omega} \frac{1}{N}\sum_{i=1}^{N} \mathrm{G}_\sigma\left(e(i)\right) \tag{5}$$

where $\widehat{W}$ denotes the optimal solution, and Ω denotes a feasible set of the parameter.

3. Maximum correntropy unscented filter

In this section, we propose to combine the MCC and a statistical linear regression model together to derive a novel non-linear filter, which can perform very well in non-Gaussian noises, since correntropy embraces second and higher order moments of the error.

Let's consider a nonlinear system described by the following equations:

$$\mathbf{x}(k) = \mathrm{f}\left(k-1, \mathbf{x}(k-1)\right) + \mathbf{q}(k-1), \tag{6}$$

$$\mathbf{y}(k) = \mathrm{h}\left(k, \mathbf{x}(k)\right) + \mathbf{r}(k). \tag{7}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ denotes a n-dimensional state vector at time step k, $\mathbf{y}(k) \in \mathbb{R}^m$ represents an m-dimensional measurement vector, f is a nonlinear system function, and h is a nonlinear measurement function and both are assumed to be continuously differentiable. The process noise $\mathbf{q}(k-1)$ and

measurement noise $\mathbf{r}(k)$ are generally assumed to meet the independence assumption with zero mean and covariance matrices

$$\mathrm{E}\left[\mathbf{q}(k-1)\mathbf{q}^T(k-1)\right] = \mathbf{Q}(k-1),\ \mathrm{E}\left[\mathbf{r}(k)\mathbf{r}^T(k)\right] = \mathbf{R}(k) \tag{8}$$

Similar to other Kalman type filters, the MCUF also includes two steps, namely the time update and measurement update:

3.1. *Time update*

A set of $2n+1$ samples, also called sigma points, are generated from the estimated state $\widehat{\mathbf{x}}(k-1|k-1)$ and covariance matrix $\mathbf{P}(k-1|k-1)$ at the last time step $k-1$:

$$\begin{aligned}
\chi^0(k-1|k-1) &= \widehat{\mathbf{x}}(k-1|k-1),\\
\chi^i(k-1|k-1) &= \widehat{\mathbf{x}}(k-1|k-1)\\
&\quad +\left(\sqrt{(n+\lambda)\mathbf{P}(k-1|k-1)}\right)_i,\ \text{for } i=1\ldots n,\\
\chi^i(k-1|k-1) &= \widehat{\mathbf{x}}(k-1|k-1)\\
&\quad -\left(\sqrt{(n+\lambda)\mathbf{P}(k-1|k-1)}\right)_{i-n},\ \text{for } i=n+1\ldots 2n.
\end{aligned} \tag{9}$$

where $\left(\sqrt{(n+\lambda)\mathbf{P}(k-1|k-1)}\right)_i$ is the i-th column of the matrix square root of $(n+\lambda)\mathbf{P}(k-1|k-1)$, with n being the state dimension and λ being a composite scaling factor, given by

$$\lambda = \alpha^2(n+\phi) - n \tag{10}$$

where α determines the spread of the sigma points, usually selected as a small positive number, and ϕ is a parameter that is often set to $3-n$.

The transformed points are then given through the process equation:

$$\chi^{i*}(k|k-1) = \mathrm{f}\left(k-1, \chi^i(k-1|k-1)\right),\ \text{for } i=0\ldots 2n \tag{11}$$

The prior state mean and covariance matrix are thus estimated by

$$\widehat{\mathbf{x}}(k|k-1) = \sum_{i=0}^{2n} w_m^i \chi^{i*}(k|k-1), \tag{12}$$

$$\begin{aligned}
\mathbf{P}(k|k-1) = &\sum_{i=0}^{2n} w_c^i \left[\chi^{i*}(k|k-1) - \widehat{\mathbf{x}}(k|k-1)\right]\\
&\times \left[\chi^{i*}(k|k-1) - \widehat{\mathbf{x}}(k|k-1)\right]^T + \mathbf{Q}(k-1).
\end{aligned} \tag{13}$$

in which the corresponding weights of the state and covariance matrix are

$$\begin{aligned} w_m^0 &= \frac{\lambda}{(n+\lambda)}, \\ w_c^0 &= \frac{\lambda}{(n+\lambda)} + (1-\alpha^2+\beta), \\ w_m^i &= w_c^i = \frac{1}{2(n+\lambda)}, \text{ for } i = 1 \dots 2n. \end{aligned} \tag{14}$$

where β is a parameter related to the prior knowledge of the distribution of $\mathbf{x}(k)$ and is set to 2 in the case of the Gaussian distribution.

3.2. *Measurement update*

Similarly, a set of $2n+1$ sigma points are generated from the prior state mean and covariance matrix

$$\begin{aligned} \chi^0(k|k-1) &= \widehat{\mathbf{x}}(k|k-1), \\ \chi^i(k|k-1) &= \widehat{\mathbf{x}}(k|k-1) \\ &\quad + \left(\sqrt{(n+\lambda)\mathbf{P}(k|k-1)}\right)_i, \text{ for } i = 1 \dots n, \\ \chi^i(k|k-1) &= \widehat{\mathbf{x}}(k|k-1) \\ &\quad - \left(\sqrt{(n+\lambda)\mathbf{P}(k|k-1)}\right)_{i-n}, \text{ for } i = n+1 \dots 2n. \end{aligned} \tag{15}$$

These points are transformed through the process equation as

$$\gamma^i(k) = \mathrm{h}(k, \chi^i(k|k-1)), \text{ for } i = 0 \dots 2n \tag{16}$$

The prior measurement mean can then be obtained as

$$\widehat{\mathbf{y}}(k) = \sum_{i=0}^{2n} w_m^i \gamma^i(k), \tag{17}$$

Further, the state-measurement cross-covariance matrix is given by

$$\mathbf{P}_{\mathbf{xy}}(k) = \sum_{i=0}^{2n} w_c^i \left[\chi^i(k|k-1) - \widehat{\mathbf{x}}(k|k-1)\right] \left[\gamma^i(k) - \widehat{\mathbf{y}}(k)\right]^T. \tag{18}$$

Next, we apply a statistical linear regression model based on the MCC to accomplish the measurement update. First, we formulate the regression model. We denote the prior estimation error of the state by

$$\eta(\mathbf{x}(k)) = \mathbf{x}(k) - \widehat{\mathbf{x}}(k|k-1) \tag{19}$$

and define the measurement slope matrix as

$$\mathbf{H}(k) = \left(\mathbf{P}^{-1}(k|k-1)\mathbf{P}_{\mathbf{xy}}(k)\right)^T \tag{20}$$

Then the measurement equation (7) can be approximated by (X. Wang et al., 2010)

$$\mathbf{y}(k) \approx \widehat{\mathbf{y}}(k) + \mathbf{H}(k)(\mathbf{x}(k) - \widehat{\mathbf{x}}(k|k-1)) + \mathbf{r}(k) \tag{21}$$

Combining (12) (17) and (21), we obtain the following statistical linear regression model:

$$\begin{bmatrix} \widehat{\mathbf{x}}(k|k-1) \\ \mathbf{y}(k) - \widehat{\mathbf{y}}(k) + \mathbf{H}(k)\widehat{\mathbf{x}}(k|k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{H}(k) \end{bmatrix} \mathbf{x}(k) + \xi(k) \tag{22}$$

where $\xi(k)$ is

$$\xi(k) = \begin{bmatrix} \eta(\mathbf{x}(k)) \\ \mathbf{r}(k) \end{bmatrix}$$

with

$$\begin{aligned} \mathbf{\Xi}(k) &= \mathrm{E}\left[\xi(k)\xi^T(k)\right] \\ &= \begin{bmatrix} \mathbf{P}(k|k-1) & 0 \\ 0 & \mathbf{R}(k) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_p(k|k-1)\mathbf{S}_p^T(k|k-1) & 0 \\ 0 & \mathbf{S}_r(k)\mathbf{S}_r^T(k) \end{bmatrix} \\ &= \mathbf{S}(k)\mathbf{S}^T(k) \end{aligned} \tag{23}$$

Here, $\mathbf{S}(k)$ can be obtained by the Cholesky decomposition of $\mathbf{\Xi}(k)$. Left multiplying both sides of (22) by $\mathbf{S}^{-1}(k)$, the statistical regression model is transformed to

$$\mathbf{D}(k) = \mathbf{W}(k)\mathbf{x}(k) + \mathbf{e}(k) \tag{24}$$

where

$$\begin{aligned} \mathbf{D}(k) &= \mathbf{S}^{-1}(k) \begin{bmatrix} \widehat{\mathbf{x}}(k|k-1) \\ \mathbf{y}(k) - \widehat{\mathbf{y}}(k) + \mathbf{H}(k)\widehat{\mathbf{x}}(k|k-1) \end{bmatrix}, \\ \mathbf{W}(k) &= \mathbf{S}^{-1}(k) \begin{bmatrix} \mathbf{I} \\ \mathbf{H}(k) \end{bmatrix}, \\ \mathbf{e}(k) &= \mathbf{S}^{-1}(k)\xi(k). \end{aligned}$$

It can be seen that $\mathrm{E}\left[\mathbf{e}(k)\mathbf{e}^T(k)\right] = \mathbf{I}$.

We are now in a position to define a cost function based on the MCC:

$$J_L\left(\mathbf{x}(k)\right) = \sum_{i=1}^{L} \mathrm{G}_\sigma\left(d_i(k) - \mathbf{w}_i(k)\mathbf{x}(k)\right) \tag{25}$$

where $d_i(k)$ is the i-th element of $\mathbf{D}(k)$, $\mathbf{w}_i(k)$ is the i-th row of $\mathbf{W}(k)$, and $L = n + m$ is the dimension of $\mathbf{D}(k)$. Under the MCC, the optimal estimate of $\mathbf{x}(k)$ arises from the following optimization:

$$\widehat{\mathbf{x}}(k) = \arg\max_{\mathbf{x}(k)} J_L\left(\mathbf{x}(k)\right) = \arg\max_{\mathbf{x}(k)} \sum_{i=1}^{L} \mathrm{G}_\sigma\left(e_i(k)\right) \tag{26}$$

where $e_i(k)$ is the i-th element of $\mathbf{e}(k)$ given by

$$e_i(k) = d_i(k) - \mathbf{w}_i(k)\mathbf{x}(k) \tag{27}$$

The optimal solution of $\mathbf{x}(k)$ can be solved through

$$\frac{\partial J_L\left(\mathbf{x}(k)\right)}{\partial \mathbf{x}(k)} = 0 \tag{28}$$

It follows easily that

$$\begin{aligned}\mathbf{x}(k) = &\left(\sum_{i=1}^{L}\left(\mathrm{G}_\sigma\left(e_i(k)\right)\mathbf{w}_i^T(k)\mathbf{w}_i(k)\right)\right)^{-1} \times \\ &\left(\sum_{i=1}^{L}\left(\mathrm{G}_\sigma\left(e_i(k)\right)\mathbf{w}_i^T(k)d_i(k)\right)\right)\end{aligned} \tag{29}$$

Since $e_i(k) = d_i(k) - \mathbf{w}_i(k)\mathbf{x}(k)$, the equation (29) is actually a fixed-point equation with respect to $\mathbf{x}(k)$ and can be rewritten as

$$\mathbf{x}(k) = \mathrm{g}\left(\mathbf{x}(k)\right) \tag{30}$$

Hence, a fixed-point iterative algorithm can be obtained as (Agarwal, Meehan, & Regan, 2001; B. Chen, Wang, Zhao, Zheng, & Principe, 2015; Singh & Principe, 2010)

$$\widehat{\mathbf{x}}(k)_{t+1} = \mathrm{g}\left(\widehat{\mathbf{x}}(k)_t\right) \tag{31}$$

with $\widehat{\mathbf{x}}(k)_t$ being the estimated state $\widehat{\mathbf{x}}(k)$ at the t-th fixed-point iteration .

The fixed-point equation (29) in matrix form can also be expressed as

$$\mathbf{x}(k) = \left(\mathbf{W}^T(k)\mathbf{C}(k)\mathbf{W}(k)\right)^{-1}\mathbf{W}^T(k)\mathbf{C}(k)\mathbf{D}(k) \tag{32}$$

where $\mathbf{C}(k) = \begin{bmatrix} \mathbf{C}_x(k) & 0 \\ 0 & \mathbf{C}_y(k) \end{bmatrix}$, with
$\mathbf{C}_x(k) = diag\left(\mathrm{G}_\sigma\left(e_1(k)\right), ..., \mathrm{G}_\sigma\left(e_n(k)\right)\right)$,
$\mathbf{C}_y(k) = diag\left(\mathrm{G}_\sigma\left(e_{n+1}(k)\right), ..., \mathrm{G}_{n+m}\left(e_{n+m}(k)\right)\right)$,
which can be further written as (see the Appendix A for a detailed derivation):

$$\mathbf{x}(k) = \widehat{\mathbf{x}}(k|k-1) + \overline{\mathbf{K}}(k)\left(\mathbf{y}(k) - \widehat{\mathbf{y}}(k)\right) \tag{33}$$

where

$$\begin{cases}\overline{\mathbf{K}}(k) = \overline{\mathbf{P}}(k|k-1)\mathbf{H}^T(k)\left(\mathbf{H}(k)\overline{\mathbf{P}}(k|k-1)\mathbf{H}^T(k) + \overline{\mathbf{R}}(k)\right)^{-1} \\ \overline{\mathbf{P}}(k|k-1) = \mathbf{S}_p(k|k-1)\mathbf{C}_x^{-1}(k)\mathbf{S}_p^T(k|k-1) \\ \overline{\mathbf{R}}(k) = \mathbf{S}_r(k)\mathbf{C}_y^{-1}(k)\mathbf{S}_r^T(k)\end{cases} \tag{34}$$

Meanwhile, the corresponding covariance matrix is updated by

$$\begin{aligned}\mathbf{P}(k|k) = &\left(\mathbf{I}-\overline{\mathbf{K}}(k)\mathbf{H}(k)\right)\mathbf{P}(k|k-1)\left(\mathbf{I}-\overline{\mathbf{K}}(k)\mathbf{H}(k)\right)^{T}\\ &+\overline{\mathbf{K}}(k)\mathbf{R}(k)\overline{\mathbf{K}}^{T}(k)\end{aligned} \tag{35}$$

Remark 1: Since $\overline{\mathbf{K}}(k)$ relies on $\overline{\mathbf{P}}(k|k-1)$ and $\overline{\mathbf{R}}(k)$, both related to $\mathbf{x}(k)$ through $\mathbf{C}_x(k)$ and $\mathbf{C}_y(k)$, respectively, the equation (33) is a fixed-point equation of $\mathbf{x}(k)$. One can solve (33) by a fixed-point iterative method. The initial value of the fixed-point iteration can be set to $\widehat{\mathbf{x}}(k|k)_0 = \widehat{\mathbf{x}}(k|k-1)$ or chosen as the least-squares solution $\widehat{\mathbf{x}}(k|k)_0 = \left(\mathbf{W}^T(k)\mathbf{W}(k)\right)^{-1}\mathbf{W}^T(k)\mathbf{D}(k)$. The value after convergence is the posterior estimate of the state $\widehat{\mathbf{x}}(k|k)$.

A detailed description of the proposed MCUF algorithm is as follows:

1) Choose a proper kernel bandwidth σ and a small positive ε; Set an initial estimate $\widehat{\mathbf{x}}(0|0)$ and corresponding covariance matrix $\mathbf{P}(0|0)$; Let $k=1$;
2) Use equations (9)$\sim$(14) to obtain prior estimate $\widehat{\mathbf{x}}(k|k-1)$ and covariance $\mathbf{P}(k|k-1)$, and calculate $\mathbf{S}_p(k|k-1)$ by Cholesky decomposition ;
3) Use (14)$\sim$(17) to compute the prior measurement $\widehat{\mathbf{y}}(k)$ and use (13) (18) and (20) to acquire the measurement slope matrix $\mathbf{H}(k)$, and construct the statistical linear regression model (22);
4) Transform (22) into (24), and let $t=1$ and $\widehat{\mathbf{x}}(k|k)_0 = \left(\mathbf{W}^T(k)\mathbf{W}(k)\right)^{-1}\mathbf{W}^T(k)\mathbf{D}(k)$;
5) Use (36)$\sim$(42) to compute $\widehat{\mathbf{x}}(k|k)_t$;

$$\widehat{\mathbf{x}}(k|k)_t = \widehat{\mathbf{x}}(k|k-1) + \widetilde{\mathbf{K}}(k)\left(\mathbf{y}(k)-\widehat{\mathbf{y}}(k)\right) \tag{36}$$

with

$$\widetilde{\mathbf{K}}(k) = \widetilde{\mathbf{P}}(k|k-1)\mathbf{H}^T(k)\Big(\mathbf{H}(k)\widetilde{\mathbf{P}}(k|k-1)\mathbf{H}^T(k)+\widetilde{\mathbf{R}}(k)\Big)^{-1}, \tag{37}$$

$$\widetilde{\mathbf{P}}(k|k-1) = \mathbf{S}_p(k|k-1)\widetilde{\mathbf{C}}_x^{-1}(k)\mathbf{S}_p^T(k|k-1), \tag{38}$$

$$\widetilde{\mathbf{R}}(k) = \mathbf{S}_r(k)\widetilde{\mathbf{C}}_y^{-1}(k)\mathbf{S}_r^T(k). \tag{39}$$

$$\widetilde{\mathbf{C}}_x(k) = diag\left(\mathrm{G}_\sigma\left(\widetilde{e}_1(k)\right),...,\mathrm{G}_\sigma\left(\widetilde{e}_n(k)\right)\right) \tag{40}$$

$$\widetilde{\mathbf{C}}_y(k) = diag\left(\mathrm{G}_\sigma\left(\widetilde{e}_{n+1}(k)\right),...,\mathrm{G}_\sigma\left(\widetilde{e}_{n+m}(k)\right)\right) \tag{41}$$

$$\widetilde{e}_i(k) = d_i(k) - \mathbf{w}_i(k)\widehat{\mathbf{x}}(k|k)_{t-1} \tag{42}$$

6) Compare the estimation at the current step and the estimation at the last step. If (43) holds, set $\widehat{\mathbf{x}}(k|k) = \widehat{\mathbf{x}}(k|k)_t$ and continue to step 7); Otherwise, $t+1 \to t$, and go back to step 5).

$$\frac{\left\|\widehat{\mathbf{x}}(k|k)_t - \widehat{\mathbf{x}}(k|k)_{t-1}\right\|}{\left\|\widehat{\mathbf{x}}(k|k)_{t-1}\right\|} \le \varepsilon \tag{43}$$

7) Update the posterior covariance matrix by (44), $k+1 \to k$ and go back to step 2).

$$\begin{aligned}\mathbf{P}(k|k) = &\left(\mathbf{I} - \widetilde{\mathbf{K}}(k)\mathbf{H}(k)\right)\mathbf{P}(k|k-1)\left(\mathbf{I} - \widetilde{\mathbf{K}}(k)\mathbf{H}(k)\right)^T \\ &+ \widetilde{\mathbf{K}}(k)\mathbf{R}(k)\widetilde{\mathbf{K}}^T(k)\end{aligned} \tag{44}$$

Remark 2: As one can see, (9)∼(18) are the unscented transformation (UT). In the MCUF, we use a statistical linear regression model and the MCC to obtain the posterior estimates of the state and covariance. With the UT and statistical linear approximation, the proposed filter can achieve a more accurate solution than the first order linearization based filters. Moreover, the usage of MCC will improve the robustness of the filter against large outliers. Usually the convergence of the fixed-point iteration to the optimal solution is very fast (see Section 4). Thus, the computational complexity of MCUF is not high. The kernel bandwidth σ, which can be set manually or optimized by trial and error methods in practical applications, is a key parameter in MCUF. In general, a smaller kernel bandwidth makes the algorithm more robust (with respect to outliers), but a too small kernel bandwidth may lead to slow convergence or even divergence of the algorithm. From (B. Chen, Wang, et al., 2015), we know that if the kernel bandwidth is larger than a certain value, the fixed-point equation (29) will surely converge to a unique fixed point. When $\sigma \to \infty$, the MCUF will obtain the least-squares solution $\left(\mathbf{W}^T(k)\mathbf{W}(k)\right)^{-1}\mathbf{W}^T(k)\mathbf{D}(k)$.

4. Illustrative examples

In this section, we present two illustrative examples to demonstrate the performance of the proposed MCUF algorithm. Moreover, the performance is measured using the following benchmarks:

$$\text{MSE}_1(k) = \frac{1}{M}\sum_{m=1}^{M}\left(x(k) - \widehat{x}(k|k)\right)^2, \text{ for } k = 1 \dots K \tag{45}$$

$$\text{MSE}_2(m) = \frac{1}{K}\sum_{k=1}^{K}\left(x(k) - \widehat{x}(k|k)\right)^2, \text{ for } m = 1 \dots M \tag{46}$$

$$\text{MSE} = \frac{1}{M}\sum_{m=1}^{M}\text{MSE}_2(m) = \frac{1}{K}\sum_{k=1}^{K}\text{MSE}_1(k) \tag{47}$$

where K is the total time steps in every Monte Carlo run and M represents the total number of Monte Carlo runs.

4.1. *Example 1*

Consider the univariate nonstationary growth model (UNGM), which is often used as a benchmark example for nonlinear filtering. The state and measurement equations are given by

$$\begin{aligned}x(k) = &0.5x(k-1) + 25\frac{x(k-1)}{1+x(k-1)^2} \\ &+ 8\cos\left(1.2(k-1)\right) + q(k-1),\end{aligned} \tag{48}$$

$$y(k) = \frac{x(k)^2}{20} + r(k). \tag{49}$$

First, we consider the case in which the noises are all Gaussian, that is,

$$\begin{aligned} q(k-1) &\sim N(0,1) \\ r(k) &\sim N(0,1) \end{aligned}$$

Table 1 lists the MSEs of x, defined in (47), and the average fixed-point iteration numbers. In the simulation, the parameters are set as $K = 500, M = 100$. Since all the noises are Gaussian, the UKF achieves the smallest MSE among all the filters. In this example, we should choose a larger kernel bandwidth in MCUF to have a good performance. One can also observe that the average iteration numbers of the MCUF are relatively small especially when the kernel bandwidth is large.

Table 1. MSEs of x and Average Iteration Numbers in Gaussian Noises.

Filter	MSE of x	Average iteration number
UKF	68.9766	—
MCUF$(\sigma = 2.0, \varepsilon = 10^{-6})$	108.9796	5.0624
MCUF$(\sigma = 3.0, \varepsilon = 10^{-6})$	94.5856	4.5431
MCUF$(\sigma = 5.0, \varepsilon = 10^{-6})$	83.7554	3.7323
MCUF$(\sigma = 8.0, \varepsilon = 10^{-6})$	86.1612	3.0431
MCUF$(\sigma = 10, \varepsilon = 10^{-6})$	85.4109	2.7919

Second, we consider the case in which the process noise is still Gaussian but the measurement noise is a heavy-tailed (impulsive) non-Gaussian noise, with a mixed-Gaussian distribution, that is,

$$\begin{aligned} q(k-1) &\sim N(0,1) \\ r(k) &\sim 0.8N(0,1) + 0.2N(0,400) \end{aligned}$$

Table 2 illustrates the corresponding MSEs of x and average iteration numbers. As one can see, in impulsive noises, when kernel bandwidth is too small or too large, the performance of MCUF will be not good. However, with a proper kernel bandwidth (say $\sigma = 2.0$), the MCUF can outperform all the filters, achieving the smallest MSE. In addition, it is evident that the larger the kernel bandwidth, the faster the convergence speed. In general, the fixed-point algorithm in MCUF will converge to the optimal solution in only few iterations.

Table 2. MSEs of x and Average Iteration Numbers in Gaussian Process Noise and Non-Gaussian Measurement Noise.

Filter	MSE of x	Average iteration number
UKF	84.3496	—
MCUF$(\sigma = 1.0, \varepsilon = 10^{-6})$	70.5870	3.5413
MCUF$(\sigma = 2.0, \varepsilon = 10^{-6})$	68.9714	3.0352
MCUF$(\sigma = 3.0, \varepsilon = 10^{-6})$	69.3548	2.7056
MCUF$(\sigma = 5.0, \varepsilon = 10^{-6})$	69.4932	2.3391
MCUF$(\sigma = 10, \varepsilon = 10^{-6})$	70.4666	2.0286

We also investigate the influence of the threshold ε on the performance. The MSEs of x and average iteration numbers with different ε (The kernel bandwidth is set at $\sigma = 2.0$) are given in Table 3. Usually, a smaller ε results in a slightly lower MSE but a larger iteration number for convergence. Without mentioned otherwise, we choose $\varepsilon = 10^{-6}$ in this work.

Table 3. MSEs of x and Average Iteration Numbers with Different ε.

Filter	MSE of x	Average iteration number
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-1}\right)$	69.7465	1.1142
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-2}\right)$	69.5853	1.3777
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-4}\right)$	69.0197	2.1753
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-6}\right)$	68.8073	3.0282
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-8}\right)$	68.8148	3.8770

Further, we consider the situation where the process and measurement noises are all non-Gaussian with mixed-Gaussian distributions:

$$q(k-1) \sim 0.8N(0, 0.1) + 0.2N(0, 10)$$
$$r(k) \sim 0.8N(0, 1) + 0.2N(0, 400)$$

With the same parameters setting as before, the results are presented in Table 4. As expected, with a proper kernel bandwidth the MCUF can achieve the best performance.

Table 4. MSEs of x and Average Iteration Numbers in Non-Gaussian Process Noise and Measurement Noise.

Filter	MSE of x	Average iteration number
UKF	84.8735	—
MCUF$\left(\sigma = 1.0, \varepsilon = 10^{-6}\right)$	71.7599	3.6104
MCUF$\left(\sigma = 2.0, \varepsilon = 10^{-6}\right)$	69.4382	3.1142
MCUF$\left(\sigma = 3.0, \varepsilon = 10^{-6}\right)$	69.8497	2.7765
MCUF$\left(\sigma = 5.0, \varepsilon = 10^{-6}\right)$	69.8505	2.3879
MCUF$\left(\sigma = 10, \varepsilon = 10^{-6}\right)$	70.0356	2.0599

4.2. *Example 2*

In this example, we consider a practical model (Simon, 2006). and the performance of EKF (Anderson & Moore, 1979), Huber-EKF (HEKF) (El-Hawary & Jing, 1995), UKF (Julier et al., 2000) and HUKF (X. Wang et al., 2010) are also presented for comparison purpose. The goal is to estimate the position, velocity and ballistic coefficient of a vertically falling body at a very high altitude. The measurements are taken by a radar system each $0.1s$. By rectangle integral in the discrete time with sampling period ΔT, we obtain the following model:

$$\begin{aligned} x_1(k_1) &= x_1(k_1-1) + \Delta T x_2(k_1-1) + q_1(k_1-1) \\ x_2(k_1) &= x_2(k_1-1) + \Delta T \rho_0 \exp\left(-x_1(k_1-1)/a\right) \\ &\quad \times x_2(k_1-1)^2 x_3(k_1-1)/2 - \Delta T g + q_2(k_1-1) \\ x_3(k_1) &= x_3(k_1-1) + q_3(k_1-1) \end{aligned} \tag{50}$$

$$y(k) = \sqrt{b^2 + \left(x_1(k) - H\right)^2} + r(k) \tag{51}$$

where the sampling time is $\Delta T = 0.001s$, that is $k = 100k_1$, the constant ρ_0 is $\rho_0 = 2$, the constant a that relates the air density with altitude is $a = 20000$, the acceleration of gravity is $g = 32.2ft/s^2$, the located altitude of radar is $H = 100000ft$, and the horizontal range between the body and the radar is $b = 100000ft$.

The state vector $\mathbf{x}(k) = \begin{bmatrix} x_1(k) & x_2(k) & x_3(k) \end{bmatrix}^T$ contains the position, velocity and ballistic coefficient. Similar to (Simon, 2006), we do not introduce any process noise in this

model. The initial state is assumed to be $\mathbf{x}(0) = \begin{bmatrix} 300000 & -20000 & 1/1000 \end{bmatrix}^T$, and the initial estimate is $\widehat{\mathbf{x}}(0|0) = \begin{bmatrix} 300000 & -20000 & 0.0009 \end{bmatrix}^T$ with covariance matrix $\mathbf{P}(0|0) = diag([1000000, 4000000, 1/1000000])$.

First, we assume that the measurement noise is Gaussian, that is,

$$\begin{aligned} &q_1(k_1) \sim 0 \\ &q_2(k_1) \sim 0 \\ &q_3(k_1) \sim 0 \\ &r(k) \sim N(0, 10000) \end{aligned}$$

In the simulation, we consider the motion during the first $50s$, and make 100 independent Monte Carlo runs, that is, $K = 500, M = 100$. Fig. 1 $\sim$ Fig. 3 show the MSE_1 (as defined in (45)) of x_1, x_2 and x_3 for different filters in Gaussian noise. The corresponding MSEs and average fixed-point iteration numbers are summarized in Table 5 and Table 6. As one can see, in this case, since the noise is Gaussian, the UKF performs the best. However, the proposed MCUF with large kernel bandwidth can outperform the other two robust Kalman type filters, namely HEKF and HUKF. One can also observe that the average iteration numbers of the MCUF are very small especially when the kernel bandwidth is large.

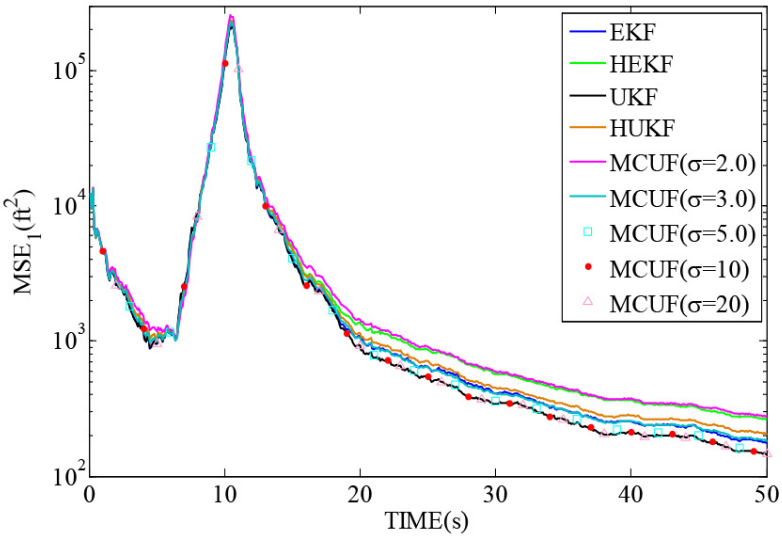

Figure 1. MSE_1 of x_1 in Gaussian noise

Table 5. MSEs of x_1, x_2 and x_3 in Gaussian Noise.

Filter	MSE of x_1	MSE of x_2	MSE of x_3
EKF	7.3254×10^3	8.6071×10^3	1.1317×10^{-8}
HEKF$(\varepsilon = 10^{-6})$	7.9266×10^3	8.9799×10^3	1.1001×10^{-8}
UKF	7.2630×10^3	8.5948×10^3	1.1308×10^{-8}
HUKF$(\varepsilon = 10^{-6})$	7.8343×10^3	8.9969×10^3	1.0983×10^{-8}
MCUF$(\sigma = 2.0, \varepsilon = 10^{-6})$	8.3984×10^3	9.4045×10^3	1.0864×10^{-8}
MCUF$(\sigma = 3.0, \varepsilon = 10^{-6})$	7.4680×10^3	8.7892×10^3	1.0865×10^{-8}
MCUF$(\sigma = 5.0, \varepsilon = 10^{-6})$	7.2943×10^3	8.6564×10^3	1.1098×10^{-8}
MCUF$(\sigma = 10, \varepsilon = 10^{-6})$	7.2674×10^3	8.6298×10^3	1.1246×10^{-8}
MCUF$(\sigma = 20, \varepsilon = 10^{-6})$	7.2642×10^3	8.6254×10^3	1.1288×10^{-8}

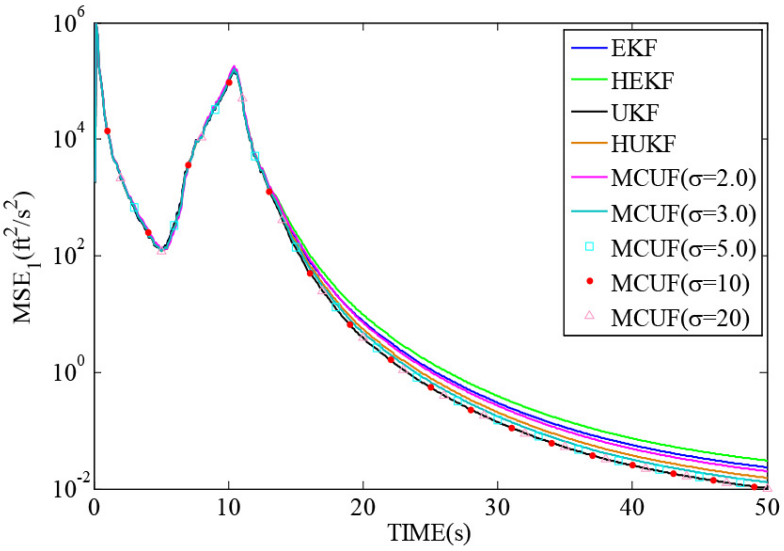

Figure 2. MSE_1 of x_2 in Gaussian noise

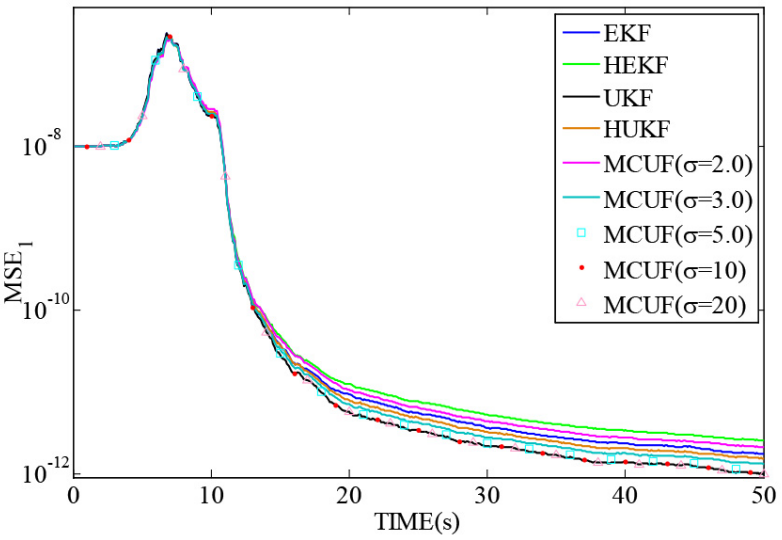

Figure 3. MSE_1 of x_3 in Gaussian noise

Table 6. Average Iteration Numbers for Every Time Step in Gaussian Noise.

Filter	Average iteration number
$\mathrm{HEKF}\left(\varepsilon = 10^{-6}\right)$	1.2342
$\mathrm{HUKF}\left(\varepsilon = 10^{-6}\right)$	1.2332
$\mathrm{MCUF}\left(\sigma = 2.0, \varepsilon = 10^{-6}\right)$	1.7881
$\mathrm{MCUF}\left(\sigma = 3.0, \varepsilon = 10^{-6}\right)$	1.5821
$\mathrm{MCUF}\left(\sigma = 5.0, \varepsilon = 10^{-6}\right)$	1.3808
$\mathrm{MCUF}\left(\sigma = 10, \varepsilon = 10^{-6}\right)$	1.1706
$\mathrm{MCUF}\left(\sigma = 20, \varepsilon = 10^{-6}\right)$	1.0521

Second, we consider the case in which the measurement noise is a heavy-tailed (impulsive) non-Gaussian noise, with a mixed-Gaussian distribution, that is,

$$
\begin{aligned}
&q_1(k_1) \sim 0 \\
&q_2(k_1) \sim 0 \\
&q_3(k_1) \sim 0 \\
&r(k) \sim 0.7N(0, 1000) + 0.3N(0, 100000)
\end{aligned}
$$

Fig. 4 $\sim$ Fig. 6 demonstrate the MSE_1 of x_1, x_2 and x_3 for different filters in non-Gaussian noise, and Table 7 and Table 8 summarize the corresponding MSEs and average iteration numbers respectively. We can see clearly that the three robust Kalman type filters (HEKF, HUKF, MCUF) are superior to their non-robust counterparts (EKF, UKF). When the kernel bandwidth is very large, the MCUF achieves almost the same performance as that of UKF. In contrast, with a smaller kernel bandwidth, the MCUF can outperform the UKF significantly. Especially, when $\sigma = 2.0$, the MCUF exhibits the smallest MSE among all the algorithms. Again, the fixed-point algorithm in MCUF will converge to the optimal solution in very few iterations.

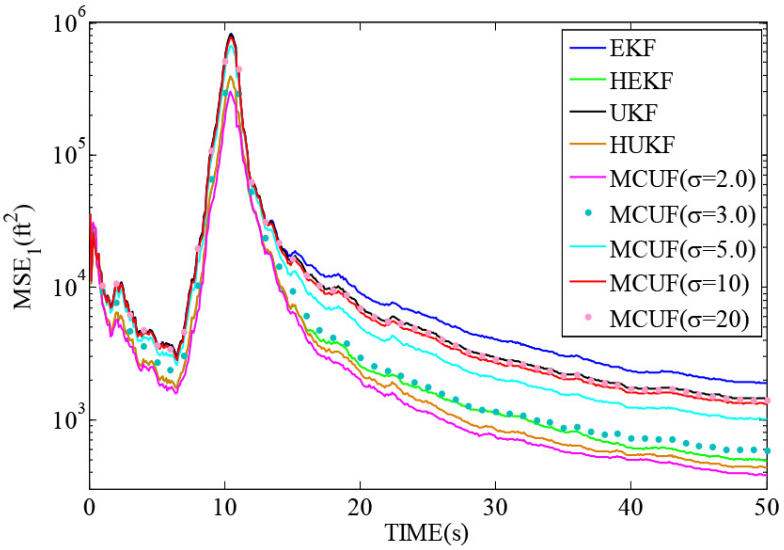

Figure 4. MSE_1 of x_1 in non-Gaussian noise

Table 7. MSEs of x_1, x_2 and x_3 in Non-Gaussian Noise.

Filter	MSE of x_1	MSE of x_2	MSE of x_3
EKF	2.9499×10^4	2.2283×10^4	1.5566×10^{-8}
HEKF$(\varepsilon = 10^{-6})$	1.4068×10^4	1.4079×10^4	8.6645×10^{-9}
UKF	2.8772×10^4	2.2331×10^4	1.5497×10^{-8}
HUKF$(\varepsilon = 10^{-6})$	1.3996×10^4	1.4212×10^4	8.6247×10^{-9}
MCUF$(\sigma = 2.0, \varepsilon = 10^{-6})$	1.1457×10^4	1.2990×10^4	7.3965×10^{-9}
MCUF$(\sigma = 3.0, \varepsilon = 10^{-6})$	1.7612×10^4	1.5970×10^4	1.0199×10^{-8}
MCUF$(\sigma = 5.0, \varepsilon = 10^{-6})$	2.3818×10^4	1.9475×10^4	1.3083×10^{-8}
MCUF$(\sigma = 10, \varepsilon = 10^{-6})$	2.7448×10^4	2.1572×10^4	1.4826×10^{-8}
MCUF$(\sigma = 20, \varepsilon = 10^{-6})$	2.8428×10^4	2.2147×10^4	1.5323×10^{-8}

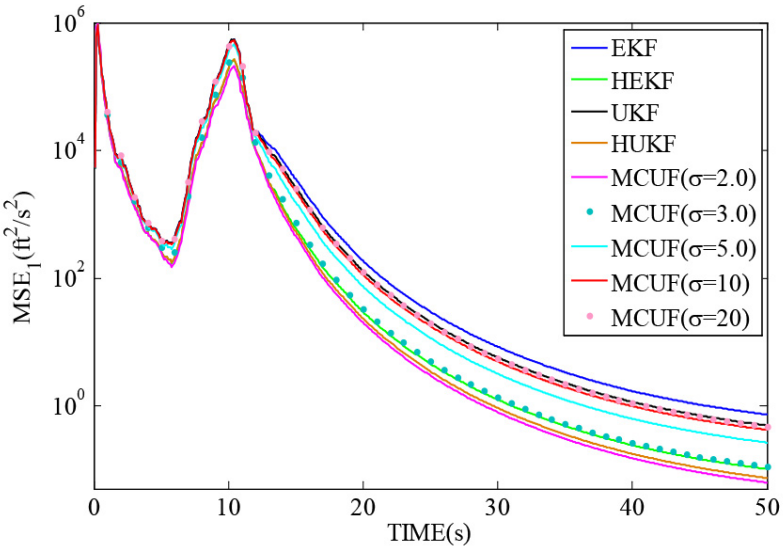

Figure 5. MSE_1 of x_2 in non-Gaussian noise

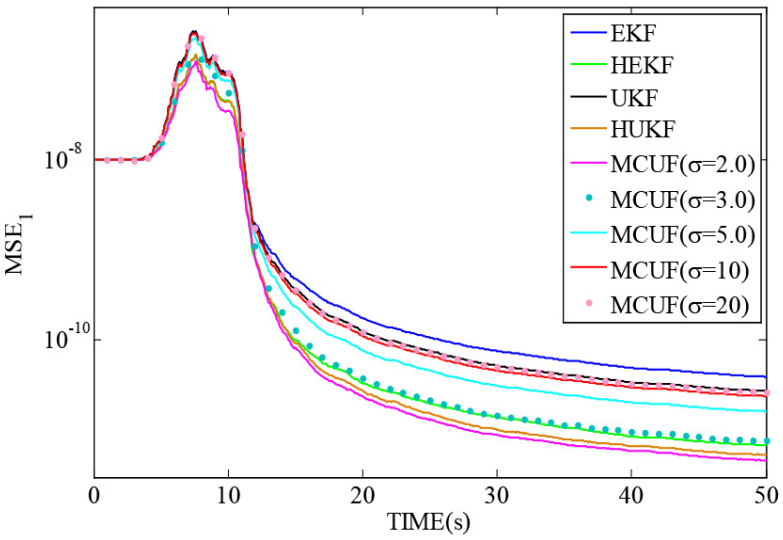

Figure 6. MSE_1 of x_3 in non-Gaussian noise

Table 8. Average Iteration Numbers for Every Time Step in Non-Gaussian Noise.

Filter	Average iteration number
$\mathrm{HEKF}\left(\varepsilon = 10^{-6}\right)$	1.2280
$\mathrm{HUKF}\left(\varepsilon = 10^{-6}\right)$	1.2300
$\mathrm{MCUF}\left(\sigma = 2.0, \varepsilon = 10^{-6}\right)$	1.4809
$\mathrm{MCUF}\left(\sigma = 3.0, \varepsilon = 10^{-6}\right)$	1.3815
$\mathrm{MCUF}\left(\sigma = 5.0, \varepsilon = 10^{-6}\right)$	1.2771
$\mathrm{MCUF}\left(\sigma = 10, \varepsilon = 10^{-6}\right)$	1.1713
$\mathrm{MCUF}\left(\sigma = 20, \varepsilon = 10^{-6}\right)$	1.0917

5. Conclusion

In this work, we propose a novel nonlinear Kalman type filter, namely the maximum correntropy unscented filter (MCUF), by using the unscented transformation (UT) to get the prior estimates of the state and covariance matrix and applying a statistical linearization regression model based on the maximum correntropy criterion (MCC) to obtain the posterior estimates (solved by a fixed-point iteration) of the state and covariance. Simulation results demonstrate that with a proper kernel bandwidth, the MCUF can achieve better performance than some existing algorithms including EKF, HEKF, UKF and HUKF particularly when the underlying system is disturbed by some impulsive noises.

Funding

This work was supported by 973 Program (No. 2015CB351703) and the National Natural Science Foundation of China (No. 61372152).

References

Agarwal, R. P., Meehan, M., & Regan, D. O. (2001). *Fixed point theory and applications.* Cambridge, U.K.: Cambridge Univ. Press.

Anderson, B., & Moore, J. (1979). *Optimal filtering.* New York, NY: Prentice-Hall.

Bessa, R. J., Miranda, V., & Gama, J. (2009). Entropy, and correntropy against minimum square error in offline, and online three-day ahead wind power forecasting. *IEEE Trans. Power Syst.*, *24*(4), 1657-1666.

Bryson, A., & Ho, Y. (1975). *Applied optimal control: Optimization, estimation and control.* Boca Raton, Florida: CRC Press.

Chen, B., Liu, X., Zhao, H., & Principe, J. C. (2015). *Maximum correntropy Kalman filter.* Retrieved from `arXiv:1509.04580`

Chen, B., & Principe, J. C. (2012). Maximum correntropy estimation is a smoothed MAP estimation. *IEEE Signal Process. Lett.*, *19*(8), 491-494.

Chen, B., Wang, J., Zhao, H., Zheng, N., & Principe, J. C. (2015). Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Process. Lett.*, *22*(10), 1723-1727.

Chen, B., Xing, L., Liang, J., Zheng, N., & Principe, J. C. (2014). Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.*, *21*(7), 880-884.

Chen, B., Xing, L., Zhao, H., Zheng, N., & Principe, J. C. (2016). Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.*, *64*(13), 3376-3387.

Chen, B., Zhu, Y., Hu, J., & Principe, J. C. (2013). *System parameter identification: Information criteria and algorithms.* Oxford, U.K.: Newnes.

Chen, X., Yang, J., Liang, J., & Ye, Q. (2012). Recursive robust least squares support vector regression based on maximum correntropy criterion. *Neurocomputing*, *97*, 63-73.

Dash, P., Hasan, S., & Panigrahi, B. (2010). Adaptive complex unscented Kalman filter for frequency estimation of time-varying signals. *IET Sci. Meas. Technol.*, *4*(2), 93-103.

El-Hawary, F., & Jing, Y. (1995). Robust regression-based EKF for tracking underwater targets. *IEEE J. Oceanic Eng.*, *20*(1), 31-41.

He, R., Hu, B., Yuan, X., & Wang, L. (2014). *Robust recognition via information theoretic learning.* Amsterdam, The Netherlands: Springer.

He, R., Hu, B., Zheng, W., & Kong, X. (2011). Robust principal component analysis based on maximum correntropy criterion. *IEEE Trans. Image Process.*, *20*(6), 1485-1494.

He, R., Zheng, W., & Hu, B. (2011). Maxinum correntropy criterion for robust face recognition. *IEEE Trans. Patt. Anal. Intell.*, *33*(8), 1561-1576.

Huber, P. (1964). Robust estimation of a location parameter. *Ann. Math. Stat.*, *35*(1), 73-101.

Julier, S., Uhlmann, J., & Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control*, *45*(3), 477-482.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME-J. Basic Eng., series D*(82), 35-45.

Karlgaard, C., & Schaub, H. (2007). Huber-based divided difference filtering. *J. Guid. Control Dynam.*, *30*(3), 885-891.

Li, D., Liu, J., Qiao, L., & Xiong, Z. (2010). Fault tolerant navigation method for satellite based on information fusion and unscented Kalman filter. *J. Syst. Eng. Electron.*, *21*(4), 682-687.

Li, L., & Xia, Y. (2013). Unscented Kalman filter over unreliable communication networks with markovian packet dropouts. *IEEE Trans. Autom. Control*, *58*(12), 3224-3230.

Liu, W., Pokharel, P. P., & Principe, J. C. (2007). Correntropy: Properties, and applications in non-gaussian signal processing. *IEEE Trans. Signal Process.*, *55*(11), 5286-5298.

Nahi, N. E. (1969). *Estimation theory and applications.* New York, NY: Wiley.

Partovibakhsh, M., & Liu, G. (2014). Adaptive unscented Kalman filter-based online slip ratio control of wheeled-mobile robot. In *Proc. 11th world congress on intell. control and autom. (WCICA)* (p. 6161-6166).

Principe, J. C. (2010). *Information theoretic learning: Renyis entropy and kernel perspectives.* New York, USA: Springer.

Schick, I., & Mitter, S. (1994). Robust recursive estimation in the presence of heavy-tailed observation noise. *Ann. Stat.*, *22*(2), 1045-1080.

Shi, L., & Lin, Y. (2014). Convex combination of adaptive filters under the maximum correntropy criterion in impulsive interference. *IEEE Signal Process. Lett.*, *21*(11), 1385-1388.

Simon, D. (2006). *Optimal state estimation: Kalman, H_∞ and nonlinear approaches.* Hoboken, NJ: A John Wiley & Sons.

Singh, A., & Principe, J. C. (2009). Using correntropy as a cost function in linear adaptive filters. In *Proc. int. joint conf. on neural networks (IJCNN)* (p. 2950-2955).

Singh, A., & Principe, J. C. (2010). A closed form recursive solution for maximum correntropy training. In *Proc. ieee int. conf. acoustics speech and signal process. (ICASSP)* (p. 2070-2073).

Wang, X., Cui, N., & Guo, J. (2010). Huber-based unscented filtering and its application to vision-based relative navigation. *IET Radar Sonar Nav.*, *4*(1), 134-141.

Wang, Y., Pan, C., Xiang, S., & Zhu, F. (2015). Robust hyperspectral unmixing with correntropy-based metric. *IEEE Trans. Image Process.*, *24*(11), 4027-4040.

Xu, J., & Principe, J. C. (2008). A pitch detector based on a generalized correlation function. *IEEE Trans. Audio Speech Lang. Process.*, *16*(8), 1420-1432.

Zhao, S., Chen, B., & Principe, J. C. (2011). Kernel adaptive filtering with maximum correntropy criterion. In *Proc. int. joint conf. on neural networks (IJCNN)* (p. 2012-2017).

Appendix A. Derivation of (33)

$$\begin{aligned}\mathbf{W}(k) &= \mathbf{S}^{-1}(k)\begin{bmatrix}\mathbf{I}\\ \mathbf{H}(k)\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{S}_p^{-1}(k|k-1) & 0\\ 0 & \mathbf{S}_r^{-1}(k)\end{bmatrix}\begin{bmatrix}\mathbf{I}\\ \mathbf{H}(k)\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{S}_p^{-1}(k|k-1)\\ \mathbf{S}_r^{-1}(k)\mathbf{H}(k)\end{bmatrix}\end{aligned} \tag{A1}$$

$$\mathbf{C}(k) = \begin{bmatrix}\mathbf{C}_x(k) & 0\\ 0 & \mathbf{C}_y(k)\end{bmatrix} \tag{A2}$$

$$\begin{aligned}\mathbf{D}\left(k\right) &= \mathbf{S}^{-1}\left(k\right)\left[\begin{array}{c}\widehat{\mathbf{x}}\left(k|k-1\right)\\ \mathbf{y}\left(k\right)-\mathrm{h}\left(k,\widehat{\mathbf{x}}(k|k-1)\right)+\mathbf{H}(k)\widehat{\mathbf{x}}(k|k-1)\end{array}\right]\\ &= \left[\begin{array}{c}\mathbf{S}_p^{-1}\left(k|k-1\right)\widehat{\mathbf{x}}\left(k|k-1\right)\\ \mathbf{S}_r^{-1}(\mathbf{y}(k)-\mathrm{h}\left(k,\widehat{\mathbf{x}}(k|k-1)\right)+\mathbf{H}(k)\widehat{\mathbf{x}}(k|k-1))\end{array}\right]\end{aligned} \tag{A3}$$

By (A1) and (A2), we have

$$\begin{aligned}&\left(\mathbf{W}^T\left(k\right)\mathbf{C}\left(k\right)\mathbf{W}\left(k\right)\right)^{-1}\\ =&\left[\left(\mathbf{S}_p^{-1}\right)^T\mathbf{C}_x\mathbf{S}_p^{-1}+\mathbf{H}^T\left(\mathbf{S}_r{}^{-1}\right)^T\mathbf{C}_y\mathbf{S}_r^{-1}\mathbf{H}\right]^{-1}\end{aligned} \tag{A4}$$

where we denote $\mathbf{S}_p\left(k|k-1\right)$ by $\mathbf{S}_p$, $\mathbf{S}_r\left(k\right)$ by $\mathbf{S}_r$, $\mathbf{C}_x\left(k\right)$ by $\mathbf{C}_x$ and $\mathbf{C}_y\left(k\right)$ by $\mathbf{C}_y$ for simplicity. Using the matrix inversion lemma with the identification:

$$\begin{aligned}&\left(\mathbf{S}_p^{-1}\right)^T\mathbf{C}_x\mathbf{S}_p^{-1}\rightarrow\mathbf{A},\ \mathbf{H}^T\rightarrow\mathbf{B},\\ &\mathbf{H}\rightarrow\mathbf{C},\ \left(\mathbf{S}_r^{-1}\right)^T\mathbf{C}_y\mathbf{S}_r^{-1}\rightarrow\mathbf{D}.\end{aligned}$$

We arrive at

$$\begin{aligned}&\left(\mathbf{W}^T\left(k\right)\mathbf{C}\left(k\right)\mathbf{W}\left(k\right)\right)^{-1}\\ =&\left(\mathbf{S}_p\mathbf{C}_x^{-1}\mathbf{S}_p^T-\mathbf{S}_p\mathbf{C}_x^{-1}\mathbf{S}_p^T\mathbf{H}^T(\mathbf{S}_r\mathbf{C}_y^{-1}\mathbf{S}_r^T+\mathbf{H}\mathbf{S}_p\mathbf{C}_x^{-1}\mathbf{S}_p^T\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{S}_p\mathbf{C}_x^{-1}\mathbf{S}_p^T\right)\end{aligned} \tag{A5}$$

Furthermore, by (A1) $\sim$ (A3), we derive

$$\begin{aligned}&\mathbf{W}^T(k)\mathbf{C}(k)\mathbf{D}(k)\\ =&\left(\mathbf{S}_p^{-1}\right)^T\mathbf{C}_x\mathbf{S}_p^{-1}\widehat{\mathbf{x}}(k|k-1)\\ &+\mathbf{H}^T\left(\mathbf{S}_r^{-1}\right)^T\mathbf{C}_y\mathbf{S}_r^{-1}\left(\mathbf{y}(k)-\mathrm{h}\left(k,\widehat{\mathbf{x}}(k|k-1)\right)+\mathbf{H}(k)\widehat{\mathbf{x}}(k|k-1)\right)\end{aligned} \tag{A6}$$

Combining (32), (A5) and (A6), we have (33).