Signal, Image and Video Processing (2024) 18 (Suppl 1):S113–S127
https://doi.org/10.1007/s11760-024-03135-y

ORIGINAL PAPER

# Robust maximum correlation entropy Kalman filtering algorithm based on S-functions under impulse noise

Yuanlian Huo[1] · Kuojian Yang[1] · Yongfeng Qi[2] · Tianci Xu[1]

Received: 13 January 2024 / Revised: 15 February 2024 / Accepted: 6 March 2024 / Published online: 8 April 2024

## Abstract

The impulse noise is heavy-tailed and distributed, which is common in many real-world scenarios of automatic control and target tracking. In order to improve the stability of Kalman filtering algorithms in the context of impulse noise, a number of Kalman filtering algorithms have been widely developed using the maximum correlation entropy instead of the minimum mean square error criterion. To better adapt to impulse noise environments. In this paper, we propose a new Kalman filtering algorithm called robust maximum correlation entropy Kalman filtering algorithm based on S-functions (HWSKF). The algorithm is based on a new cost function framework. On the one hand, this framework employs the properties of the Softplus function fused with the maximum correlation entropy to ensure that the algorithm is better adapted to the impulsive noise environment with faster convergence speed and stronger steady-state properties. On the other hand, the introduction of Huber regression weighting can effectively exclude the effect of outliers in the case of non-Gaussian noise. Under different impulse noise environments, we demonstrate through Monte Carlo simulation experiments that the algorithm proposed in this paper possesses high stationarity compared with the classical Kalman filtering algorithm (KF), the maximum correlation entropy Kalman filtering algorithm (MCKF), and the maximum correlation entropy Kalman filtering algorithm based on the student's kernel (STTKF).

**Keywords** Kalman filter algorithm · Huber regression · Impulse noise · Softplus function · Robust estimation

## 1 Introduction

The Kalman filter is an efficient recursive filter [1]. Because of its excellent mathematical properties and good performance, it has been widely used in many fields such as navigation systems [2], vehicle positioning [3, 4], target tracking [5], communication systems to battery management systems [6, 7].

### 1.1 Kalman filtering in information learning theory

KF is proposed based on the minimum mean square error (MMSE) criterion [8], whose use of the MMSE criterion minimizes the mean squared error between the estimated and true values [9, 10]. Thus, it can show good performance in a Gaussian noise environment [11]. However, the performance is severely degraded in real-world environments, which usually face the effects of impulse noise. For this reason, researchers have focused on Kalman filtering algorithms that can be applied to non-Gaussian noise environments. In information-theoretic learning (ITL), correlation entropy is used as a local similarity measure that focuses only on the local portion of the probability density in the kernel size range, making it a robust cost in signal processing and machine learning. Literature [12] proposed a maximum correlation entropy Kalman filtering algorithm (MCKF), that successfully reduces the effect of non-Gaussian noise by exploiting the insensitivity

✉ Yuanlian Huo
huoyuanlian@163.com

Kuojian Yang
532152961@qq.com

Yongfeng Qi
qiyf@nwnu.edu.cn

Tianci Xu
1620207634@qq.com

[1] College of Physics and Electronic Engineering, Northwest Normal University, 967 Anning East Road, Lanzhou 730070, Gansu, China

[2] College of Computer Science and Engineering, Northwest Normal University, 967 Anning East Road, Lanzhou 730070, Gansu, China

of a local metric like correlation entropy to outliers. Since then a series of Kalman filtering algorithms based on maximum correlation entropy have been proposed to improve the system's steady state. For example, to solve the dispersion-prone property of the small kernel width state, the maximum correlation entropy Kalman filtering algorithm (m-MCKF) based on m-estimation has been proposed in the literature [13]. To find a suitable noise covariance matrix, an adaptive maximum correlation entropy Kalman filtering algorithm (AMCKF) is proposed in literature [14]. Nevertheless, MCC is usually implemented based on a Gaussian kernel function, researchers found that the performance of the algorithm can be optimized by changing the kernel function, so literature [15] proposed a maximum correlation entropy Kalman filtering algorithm based on student's kernel (STTKF), and literature [16] proposed a Kalman filtering algorithm based on Cauchy kernel (CKKF).

However, MCC can fix the peak of the probability density function to zero, which makes the algorithm accuracy limited, for this reason, researchers take advantage of this advantage of minimum error entropy (MEE) to optimize directly for the distribution of errors and propose the Kalman filtering algorithm (MEE-KF) based on minimum error entropy in literature [17]. Although the MEE criterion reduces the uncertainty of the error variable, it does not converge the error to zero due to the translational invariance of the entropy; for this reason, a central error entropy Kalman filtering algorithm (CEEKF) has been proposed in the literature [18]. Literature [19] proposes an outlier-robust Kalman filters with mixture correntropy. To solve the problem that the shape of the kernel function of the error entropy cannot be changed freely, the researchers make use of the property that the generalized Gaussian kernel function can be adjusted to the shape freely and propose a Kalman filtering algorithm based on the generalized minimum error entropy criterion (GMEEKF) in the literature [20]. Since then to improve the robust performance of the algorithm at the expense of algorithm complexity, literature [21]-based adaptive kernel sensitive loss Kalman filtering algorithm (AKRSLKF) and various other Kalman filtering algorithms have been proposed.

### 1.2 Research motivation and contributions

In the course of our research, we found that compared to some other cost functions, the Softplus function has a better tolerance to outliers, which makes the model robust to handling noisy data. Furthermore, the smoothness of Softplus will enable easier finding of optimal solutions when using optimization algorithms such as gradient descent. M-estimation as a robust estimation method [22, 23], on the one hand, his score function and its derivative are bounded, so the influence function is bounded and M-estimation is robust [24–26]. On the other hand, the objective function of M-estimation can be freely chosen within a certain range to accommodate different needs, and the slower the objective function grows as the absolute value of the residuals increases, the more robust the estimation becomes.

Therefore, in this paper, a robust maximum correlation entropy Kalman filtering algorithm based on S-function is developed to solve the problem of insufficient robustness and convergence speed of maximum correlation entropy to deal with impulse noise. The algorithm first exploits the smooth robustness of S-shaped functions and the insensitivity of the maximum correlation entropy local measure to outliers, makes it based on the maximum correlation entropy Kalman filtering algorithm, which speeds up the convergence and enhances the robustness to a lesser extent. However, with the algorithm at large kernel widths, we find that the robust performance of the algorithm has some magnitude of degradation compared to MCKF. For that, we introduce the robustness method of Huber regression weighting so that it can better handle the effects of outliers, making the algorithm more robust here while having a faster convergence rate. Finally, the Monte Carlo simulation experiments in different impulse noise cases prove that the algorithms in this paper have faster convergence speed and better robust performance compared with KF, MCKF, STTKF and other algorithms.

### 1.3 Structure of the article

The structure of this paper is organized as follows: Section 2 introduces the classical Kalman filtering algorithm and the correlation entropy theory, Sect. 3 carries out the derivation of the Kalman filtering algorithm in this paper, and then the mean square error analysis and the time complexity analysis are given in Sects. 4 and 5, we carry out the simulation comparison in the Gaussian as well as non-Gaussian environments to make use of Monte Carlo experiments of the two models to illustrate the superiority of the algorithm more effectively, and finally, the conclusion is given in Sect. 6.

## 2 Classical Kalman filtering algorithm and correlation entropy

### 2.1 Classical Kalman filter algorithm

The Kalman filtering algorithm acts as an optimal estimator under the assumption of linearity and Gaussian, and the linear system is defined as the following state and measurement equations.

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{1}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ represents the $n$-dimensional state vector at moment, represents the $m$-dimensional observation vector at moment $k$, $\mathbf{A}_{k-1}$ and $\mathbf{C}_k$ represent the state transfer matrix and the observation matrix, $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ represent the process noise vector and the observation noise vector, which are uncorrelated and have a mean value of zero. Define the corresponding covariance matrices as $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ denoted by

$$E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^{\mathrm{T}}] = \mathbf{Q}_{k-1} \tag{3}$$

$$E[\mathbf{v}_k\mathbf{v}_k^{\mathrm{T}}] = \mathbf{R}_k. \tag{4}$$

In general, the Kalman filtering algorithm can represented through two steps: prediction and update:

(1) Prediction: Obtain a priori state estimation matrix and a priori error covariance matrix

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{5}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{6}$$

(2) Update: Obtain Kalman gain, a posteriori state estimation matrix, and a posteriori covariance matrix

$$\mathbf{K} = \mathbf{P}_{k|k-1}\mathbf{C}_k^{\mathrm{T}}(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1} \tag{7}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}) \tag{8}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}\mathbf{C}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}\mathbf{C}_k)^{\mathrm{T}} + \mathbf{K}\mathbf{R}_k\mathbf{K}^{\mathrm{T}} \tag{9}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the unit matrix for $n \times n$.

### 2.2 Correlation entropy

Define two random variables $X, Y$ with joint probability density $F_{XY}(x, y)$ and correlation entropy criterion defined as

$$V(X, Y) = E[\alpha(X, Y)] = \int \alpha(X, Y)d(F_{XY}(x, y)) \tag{10}$$

where $E[\cdot]$ is the mathematical expectation and $\alpha(\cdot, \cdot)$ is the kernel function, denoted as

$$\alpha(X, Y) = G_\sigma(e) = \exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{11}$$

where $e = X - Y$, $\sigma$ is the kernel width and $\sigma > 0$. We can obtain the cost function of the maximum correlation entropy Kalman filtering algorithm as

$$J_{MCC}(\mathbf{x_k}) = \frac{1}{L}\sum_{i=1}^{L} G_\sigma(e) \tag{12}$$

where $L = n + m$.

## 3 Algorithm proposal

### 3.1 Enhanced models

First rewrite Eq. (1) and (2) as

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{C}_k \end{bmatrix}\mathbf{x}_k + o_k \tag{13}$$

where $o_k = \begin{bmatrix} -\left(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k|k-1}\right) \\ \mathbf{v}_k \end{bmatrix}$,
with

$$E\left[o_k o_k^{\mathrm{T}}\right] = \begin{bmatrix} \mathbf{P}_{k|k-1} & 0 \\ 0 & \mathbf{R}_k \end{bmatrix} = \begin{bmatrix} \upsilon_{P:k|k-1}\upsilon_{P:k|k-1}^{\mathrm{T}} & 0 \\ 0 & \upsilon_{R:k}\upsilon_{R:k}^{\mathrm{T}} \end{bmatrix} = \upsilon_k\upsilon_k^{\mathrm{T}} \tag{14}$$

where $\upsilon_{P:k|k-1}$, $\upsilon_{R:k}$ and $\upsilon_k$ are obtained from $\mathbf{P}_{k|k-1}$, $\mathbf{R}_k$ and $E\left[o_k o_k^{\mathrm{T}}\right]$ by Cholesky decomposition, respectively.

Simultaneous left-multiplication of $\upsilon_k^{-1}$ on both sides of Eq. (13) gives us

$$\mathbf{D}_k = \mathbf{W}_k\mathbf{x}_k + \mathbf{e}_k \tag{15}$$

where $\mathbf{D}_k = \upsilon_k^{-1}\begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{y}_k \end{bmatrix}$, $\mathbf{W}_k = \upsilon_k^{-1}\begin{bmatrix} \mathbf{I} \\ \mathbf{C}_k \end{bmatrix}$ and $\mathbf{e}_k = \upsilon_k^{-1}\begin{bmatrix} -\left(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k|k-1}\right) \\ \mathbf{v}_k \end{bmatrix}$. It can be further written as $\mathbf{D}_k = [d_{1:k}, d_{2:k}, \cdots, d_{L:k}]^{\mathrm{T}}$, $\mathbf{W}_k = [\mathbf{w}_{1:k}, \mathbf{w}_{2:k}, \cdots, \mathbf{w}_{L:k}]^{\mathrm{T}}$ and $\mathbf{e}_k = [e_{1:k}, e_{2:k}, \cdots, e_{L:k}]^{\mathrm{T}}$. Thus

$$e_{i:k} = d_{i:k} - \mathbf{w}_{i:k}\mathbf{x}_k. \tag{16}$$

### 3.2 Derivation of HWSKF

To increase the convergence speed and steady state of the algorithm based on the MCKF algorithm and better adapt to the effect of impulse noise, the HWSKF algorithm presented in this section generates a new cost function framework that combines the Softplus function with the MCKF algorithm. Observe the expression for the Softplus function, which has the derivative form $\frac{1}{1+e^{-x}}$, such a function is consistent with the outer manifestation of filter theory. The Softplus function expression is in logarithmic form, a feature that makes the algorithm faster to compute. On the other hand, Huber estimation is a robust statistical method designed to deal with outliers or outliers in regression analysis, while maintaining efficiency and accuracy for non-outliers. At its core is a loss function that combines the characteristics of squared loss (used in least squares) and absolute value loss (used in L1 regularization), so it can give optimal results when the noise obeys a normal distribution, and when it does not conform to a normal distribution, it can effectively address the effects of outliers and get better results than least squares.

In summary, a new Huber-weighted S-type fusion cost function is defined as

$$J_L(\mathbf{x}_k) = \frac{1}{L}\sum_{i=1}^{L} \vartheta_{i:k} \log\left(1 + \exp^{-\frac{{e_i}^2}{2\sigma^2}}\right) \tag{17}$$

where $\vartheta_i(x; e_i) = \begin{cases} 1, |e_i| \le \delta \\ \frac{\delta}{|e_i|}, |e_i| > \delta \end{cases}$, $\vartheta_i$ is the weighting factor for the $i$th point. Thus, the optimal estimate of $\mathbf{x}_k$ under this condition is

$$\hat{\mathbf{x}}_k = \underset{\mathbf{x}_k}{\arg\max}\, J_L(\mathbf{x}_k) = \underset{\mathbf{x}_k}{\arg\max}\, \vartheta_{i:k} \log\left(1 + \exp^{-\frac{{e_i}^2}{2\sigma^2}}\right) \tag{18}$$

where $e_i$ is the $i$-th element of $\mathbf{e}_k$.

Therefore, the optimal estimate $\mathbf{x}_k$ can be computed as

$$\frac{\partial J_L(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \sum_{i=1}^{L}\left[\vartheta_{i:k}\frac{\exp^{-\frac{{e_i}^2}{2\sigma^2}}}{1+\exp^{-\frac{{e_i}^2}{2\sigma^2}}}\mathbf{w}_{i:k}^{\mathrm{T}}(d_{i:k} - \mathbf{w}_{i:k}\mathbf{x}_k)\right] = 0 \tag{19}$$

with

$$\mathbf{x}_k = \left(\sum_{i=1}^{L}[\vartheta_{i:k}\frac{\exp^{-\frac{{e_i}^2}{2\sigma^2}}}{1+\exp^{-\frac{{e_i}^2}{2\sigma^2}}}\mathbf{w}_{i:k}^{\mathrm{T}}\mathbf{w}_{i:k}]\right)^{-1} \left(\sum_{i=1}^{L}[\vartheta_{i:k}\frac{\exp^{-\frac{{e_i}^2}{2\sigma^2}}}{1+\exp^{-\frac{{e_i}^2}{2\sigma^2}}}\mathbf{w}_{i:k}^{\mathrm{T}}d_{i:k}]\right) \tag{20}$$

Equation (20) is essentially an iteration of an indefinite point that can be rewritten as

$$\mathbf{x}_k = f(\mathbf{x}_k) \tag{21}$$

with

$$f(\mathbf{x}_k) = \left(\sum_{i=1}^{L}[\vartheta_{i:k}\frac{\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}{1+\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}\mathbf{w}_{i:k}^{\mathrm{T}}\mathbf{w}_{i:k}]\right)^{-1} \times \left(\sum_{i=1}^{L}[\vartheta_{i:k}\frac{\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}{1+\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}\mathbf{w}_{i:k}^{\mathrm{T}}d_{i:k}]\right) \tag{22}$$

An indefinite point iteration can be obtained relatively easily

$$\hat{\mathbf{x}}_{k:t+1} = f(\hat{\mathbf{x}}_{k:t}) \tag{23}$$

where $\hat{\mathbf{x}}_{k:t+1}$ denotes the solution at iteration $t+1$ of the indefinite point.

Equation (23) can be re-expressed as

$$\mathbf{x}_k = (\mathbf{W}_k^{\mathrm{T}}\Omega_k\mathbf{W}_k)^{-1}\mathbf{W}_k^{\mathrm{T}}\Omega_k\mathbf{D}_k \tag{24}$$

where $\Omega_k = \begin{bmatrix} \Omega_{P:k} & 0 \\ 0 & \Omega_{R:k} \end{bmatrix}$, with

$$\Omega_{P:k} = diag(\vartheta_{1:k}A_{1:k}, \vartheta_{2:k}A_{2:k}, \cdots, \vartheta_{n:k}A_{n:k}) \tag{25}$$

$$\Omega_{R:k} = diag(\vartheta_{n+1:k}A_{n+1:k}, \vartheta_{n+2:k}A_{n+2:k}, \cdots, \vartheta_{n+m:k}A_{n+m:k}) \tag{26}$$

where $A_{L:k} = \vartheta_{L:k}\frac{\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}{1+\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}$.

Combining Eq. (8) and (24), by means of the matrix inversion theorem, it is obtained that

$$\tilde{\mathbf{K}} = \tilde{\mathbf{P}}_{k|k-1}\mathbf{C}_k^{\mathrm{T}}(\mathbf{C}_k\tilde{\mathbf{P}}_{k|k-1}\mathbf{C}_k^{\mathrm{T}} + \tilde{\mathbf{R}}_k)^{-1} \tag{27}$$

$$\tilde{\mathbf{P}}_{k|k-1} = \upsilon_{P:k|k-1}\Omega_{P:k}^{-1}\upsilon_{P:k|k-1}^{\mathrm{T}} \tag{28}$$

$$\tilde{\mathbf{R}}_k = \upsilon_{R:k}\Omega_{R:k}^{-1}\upsilon_R^{\mathrm{T}} \tag{29}$$

In summary, the algorithm flow of this paper is as follows

Step 0, choose a small positive number $\iota$, set the initial state estimate $\mathbf{x}_{0|0}$ and the initial error covariance matrix $\mathbf{P}_{0|0}$.

Step 1, the a priori state estimation matrix $\hat{\mathbf{x}}_{k|k-1}$ and the a priori error covariance matrix $\mathbf{P}_{k|k-1}$ are obtained from Eq. (5) and (6), and $\upsilon_k$ is obtained by positive definite decomposition.

Step 2, Let $t = 1$ and $\hat{\mathbf{x}}_{k|k:0} = \hat{\mathbf{x}}_{k|k-1}$, perform indefinite iteration.

Step 3, using Eqs. (30)–(38), was able to obtain $\hat{\mathbf{x}}_{k|k:t}$.

$$\hat{\mathbf{x}}_{k|k:t} = \hat{\mathbf{x}}_{k|k-1} + \tilde{\mathbf{K}}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}) \tag{30}$$

with

$$\tilde{\mathbf{K}} = \tilde{\mathbf{P}}_{k|k-1}\mathbf{C}_k^{\mathrm{T}}(\mathbf{C}_k\tilde{\mathbf{P}}_{k|k-1}\mathbf{C}_k^{\mathrm{T}} + \tilde{\mathbf{R}}_k)^{-1} \tag{31}$$

$$\tilde{\mathbf{P}}_{k|k-1} = \upsilon_{P:k|k-1}\Omega_{P:k}^{-1}\upsilon_{P:k|k-1}^{\mathrm{T}} \tag{32}$$

$$\tilde{\mathbf{R}}_k = \upsilon_{R:k}\Omega_{R:k}^{-1}\upsilon_R^{\mathrm{T}} \tag{33}$$

$$\Omega_{P:k} = diag(\vartheta_{1:k}A_{1:k}, \vartheta_{2:k}A_{2:k}, \cdots, \vartheta_{n:k}A_{n:k}) \tag{34}$$

$$\Omega_{R:k} = diag(\vartheta_{n+1:k}A_{n+1:k}, \vartheta_{n+2:k}A_{n+2:k}, \cdots, \vartheta_{n+m:k}A_{n+m:k}) \tag{35}$$

$$A_{L:k} = \vartheta_{L:k} \frac{\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}}{1+\exp^{-\frac{(d_{i:k}-\mathbf{w}_{i:k}\mathbf{x}_k)^2}{2\sigma^2}}} \tag{36}$$

Step 4, Comparison utilizes Eq. (37) to compare the estimates of the current state and the previous state, if satisfied then proceed to the next step and vice versa return to Step 3.

$$\frac{||\hat{\mathbf{x}}_{k|k:t} - \hat{\mathbf{x}}_{k|k:t-1}||}{||\hat{\mathbf{x}}_{k|k:t}||} \le \iota \tag{37}$$

Step 5, update the posterior error covariance matrix and return to Step 1.

$$\mathbf{P}_{k|k} = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{P}_{k|k-1}(I - \tilde{\mathbf{K}}\mathbf{C}_k)^{\mathrm{T}} + \tilde{\mathbf{K}}\mathbf{R}_k\tilde{\mathbf{K}}^{\mathrm{T}} \tag{38}$$

## 4 Theoretical analysis

In this section, the performance of the proposed algorithm is theoretically analyzed in terms of mean square error as well as time complexity.

### 4.1 Mean error analysis

The estimation error of the state vector we can express in Eq. (39)

$$\gamma_k = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}. \tag{39}$$

It can be rewritten through Eq. (8) as

$$\gamma_k = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{A}_k\gamma_{k-1} + (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{w}_k - \tilde{\mathbf{K}}\mathbf{v}_k. \tag{40}$$

Therefore, the statistical expectation of $\gamma_k$ is

$$\begin{aligned} E[\gamma_k] &= (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{A}_k E[\gamma_{k-1}] \\ &= (\mathbf{I} - [\tilde{\mathbf{P}}_{k|k-1}^{-1} + \mathbf{C}_k^{\mathrm{T}}\tilde{\mathbf{R}}_k^{-1}\mathbf{C}_k]^{-1}\mathbf{C}_k^{\mathrm{T}}\tilde{\mathbf{R}}_k^{-1}\mathbf{C}_k) \\ &\quad \times \mathbf{A}_k E[\gamma_{k-1}] \end{aligned} \tag{41}$$

Assuming that the matrix $\mathbf{A}_k$ is stable, $\tilde{\mathbf{P}}_{k|k-1}$ and $\mathbf{C}_k^{\mathrm{T}}\tilde{\mathbf{R}}_k^{-1}\mathbf{C}_k$ are positive definite or semi-positive definite. So it can be concluded that is stable. Therefore, we can conclude that the algorithm will work unbiased when the state matrix is stable.

### 4.2 Mean square error analysis

Considering the errors in Eqs. (34) and (35), the covariance matrix of the error can be rewritten as

$$\begin{aligned} E[\gamma_k\gamma_k^{\mathrm{T}}] = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{A}_k E[\gamma_{k-1}\gamma_{k-1}^{\mathrm{T}}]\mathbf{A}_k^{\mathrm{T}}(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)^{\mathrm{T}} \\ +(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{Q}_k(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)^{\mathrm{T}} + \tilde{\mathbf{K}}\mathbf{R}_k\tilde{\mathbf{K}}^{\mathrm{T}} \end{aligned} \tag{42}$$

Suppose that $\mathbf{C}_k$, $\mathbf{A}_k$, $\mathbf{Q}_k$, and $\mathbf{R}_k$ are all time-invariant and that $(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{Q}_k(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)^{\mathrm{T}} + \tilde{\mathbf{K}}\mathbf{R}_k\tilde{\mathbf{K}}^{\mathrm{T}}$ and $(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{A}_k$ are time-invariant. Therefore, we can conclude that $E[\gamma_k\gamma_k^{\mathrm{T}}]$ converges. Furthermore, assuming that

$$\begin{aligned} &\lim_{k\to\infty} \varsigma_k = \varsigma \\ &\lim_{k\to\infty} \psi_{\mathbf{k}} = \psi \\ &\lim_{k\to\infty} \varphi_k = \varphi \end{aligned} \tag{43}$$

where $\varsigma_k = E[\gamma_k\gamma_k^{\mathrm{T}}]$,$\psi_k = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{A}_k$ and $\varphi_k = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)\mathbf{Q}_k(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{C}_k)^{\mathrm{T}} + \tilde{\mathbf{K}}\mathbf{R}_k\tilde{\mathbf{K}}^{\mathrm{T}}$.

Then, we have

$$\varsigma = \psi\varsigma\psi + \varphi \tag{44}$$

Eventually, we can get the closed solution from Eq. (44)

$$Vec(\varsigma) = (\mathbf{I} - \psi \otimes \psi)^{-1} Vec(\varphi) \tag{45}$$

where ⊗ is the Kronecker product.

### 4.3 Time complexity analysis

Next, we analyze the time complexity of the algorithm in Table 1 and give the computational complexity of the basic equations.

For the classical Kalman filtering algorithm involved in Eqs. (5)–(9), the computational complexity is

$$\tau_{KF} = 8n^3 + 10n^2m - n^2 + 6nm^2 - n + o(m^3) \tag{46}$$

And for MCKF the formulas involved are Eqs. (5), (6), (16), (30)–(33), (38) and two Gaussian kernel formulas, which define the average number of indefinite point iterations as $T$, so the computational complexity of MCKF is

$$\begin{aligned} \tau_{MCKF} = &(2T+8)n^3 + (4T+6)n^2m + (2T-1)n^2 \\ &+(4T+2)nm^2 + (3T-1)nm + (4T-1)n \\ &+2Tm^3 + 2Tm + To(n^3) + 2To(m^3) \end{aligned} \tag{47}$$

The formulas involved in the algorithm of this paper are Eqs. (5), (6), (16), (30)–(35), and (38), the same as defined above the average number of indefinite point iterations as $T$, so the computational complexity of the algorithm of this paper is shown in Eq. (48)

$$\begin{aligned} \tau_{HSWKF} = &(2T+8)n^3 + (4T+6)n^2m \\ &+(4T+2)nm^2 + (T+3)nm \end{aligned}$$

**Table 1** Time complexity analysis

| Equation | Addition and multiplication | Division, matrix inversion, chemical decomposition, and exponentiation |
|---|---|---|
| (5) | $2n^2 - n$ | 0 |
| (6) | $4n^3 - n^2$ | 0 |
| (7) | $4n^2m + 4nm^2 - 3nm$ | $o(m^3)$ |
| (8) | $4nm$ | 0 |
| (9) | $4n^3 + 6n^2m - 2n^2 + 2nm^2 - nm$ | 0 |
| (16) | $2n$ | 0 |
| (30) | $4nm$ | 0 |
| (31) | $4n^2m + 4nm^2 - 3nm$ | $o(m^3)$ |
| (32) | $2n^3$ | $n + o(n^3)$ |
| (33) | $2m^3$ | $m + o(m^3)$ |
| (34) | $4n^2 + 2n$ | $3n$ |
| (35) | $4nm + 2m$ | $3m$ |
| (38) | $4n^3 + 6n^2m - 2n^2 + 2nm^2 - nm$ | 0 |

$$+(4T-1)n^2 + (6T+1)n + 2Tm^3 + 6Tm$$
$$+To(n^3) + 2To(m^3) \tag{48}$$

Remark: the number of indefinite point iterations is relatively small, compared with KF, and the time complexity of this paper is moderate, but compared with MCKF, the time complexity of the algorithm in this paper is not much different.

## 5 Simulation experiment

To demonstrate the superiority of the proposed algorithm, over 100 Monte Carlo simulations are performed using two models, which are run on MATLAB version 2021a running on an i5-7500 CPU. We used 1000 samples in each run to measure the mean square deviation of the algorithm, which is shown in Eq. (49).

$$MSD = 20\log_{10}||x_k - \hat{x}_{k|k}||^2 \tag{49}$$

In simulation experiments before we define the model fixed parameters, for the process noise are used Gaussian distribution of the case, and the measurement noise is divided into a Gaussian distribution $q(k-1) \sim N(0, 0.01)$ noise and three different impulse noise two cases. The noise figure is shown in Fig. 1.

$$\begin{cases} r_1(k) \sim N(0, 0.01) \\ r_2(k) = M(0.9, 0, 0.01, 10) \\ r_3(k) = M(0.9, 0, 0.01, 20) \\ r_4(k) = M(0.9, 0, 0.01, 30) \end{cases} \tag{50}$$

### 5.1 Simulation 1

First, we consider a simple linear model literature [3] with the following state and observation equations

$$\mathbf{x}_k = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{51}$$

$$\mathbf{y}_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \tag{52}$$

where $\mathbf{x}_k$ is a two-dimensional column vector, $\theta = {}^{18}/_{\pi}$, the threshold is set to $\iota = 10^{-4}$, and $\mathbf{x}_0$, $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ are set as follows.

$$\begin{cases} \mathbf{x}_0 \sim N(0, \mathbf{I}_n) \\ \hat{\mathbf{x}}_{0|0} \sim N(\mathbf{x}_0, \mathbf{I}_n) \\ \mathbf{P}_{0|0} = \mathbf{I}_n \end{cases} \tag{53}$$

We compare the performance of this algorithm with KF, MCKF and STTKF for these four noise cases. In the MCKF algorithm, we have chosen $\sigma = 2, 3, 5$ three kernel width cases for simulation. For the STTKF algorithm, we similarly choose $\sigma = 2, 3, 5(v = 3)$ three identical comparison parameters for the simulation. In addition, to reflect the strong robustness of the algorithm, we compare the algorithm in this paper with the S-type maximum correlation entropy Kalman filtering algorithm without the addition of Huber's regression weighting, which we name WSKF, and the function cost is $J_{WSKF}(\mathbf{x}_k) = \frac{1}{L}\sum_{i=1}^{L}\log(1 + \exp^{-\frac{e_i{}^2}{2\sigma^2}})$. The kernel widths of the WSKF are also selected for the $\sigma = 2, 3, 5$ three cases.

We perform simulations under Gaussian noise and three different impulse noise shapes. The MSD plots in Gaussian noise environment we placed in Fig. 2, and the simulations

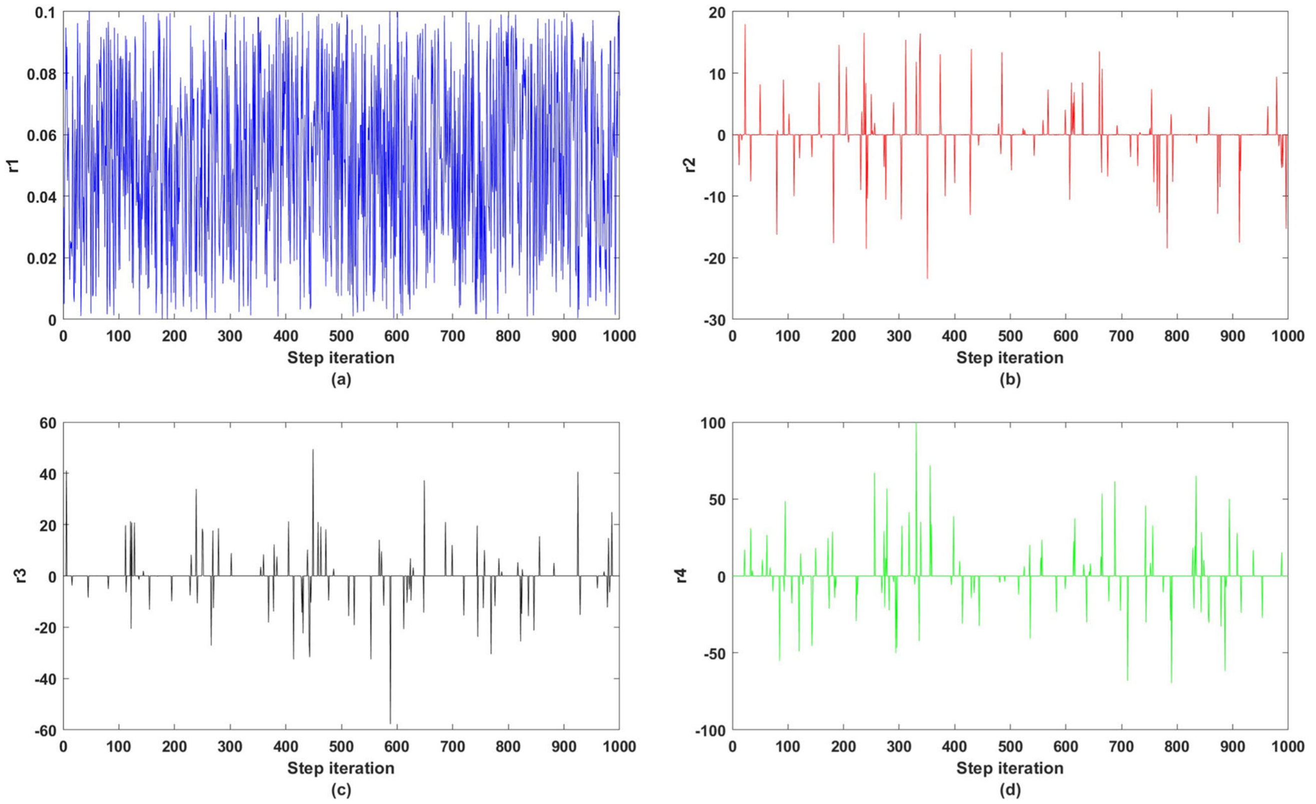
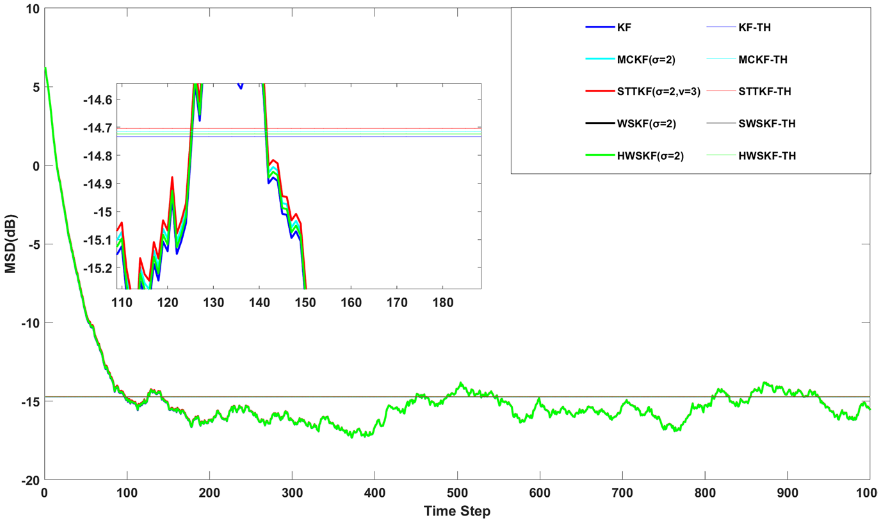


**Fig. 1** Four noise maps. **a** Case 1 Gaussian noise map; **b** case 2 impulsive noise map; **c** case 3 impulsive noise map; **d** case 4 impulsive noise map

**Fig. 2** Comparison of the algorithms under Gaussian noise in Example 1



in different pulse environments are shown in Figs. 3, 4 and 5, where KF-TH, MCKF-TH, STTKF-TH,WSKF-TH, and HWSKF-TH, denote the theoretical steady state errors of KF, MCKF, STTKF, WSKF, and HWSKF, respectively. We can see in Fig. 1 that KF has the lowest mean square deviation, the fastest convergence and the best performance when the system is in a Gaussian noise environment. However, when the system is under the influence of impulse noise, Figs. 2, 3 and 4 clearly show that the classical Kalman loses its original steady state properties, which are weakened with the enhancement of impulse noise. But whatever the intensity of the impulse noise environment, the proposed algorithm outperforms the classical Kalman filtering algorithm, the maximum correlation entropy Kalman filtering algorithm, and the maximum correlation entropy Kalman filtering algorithm based on the learning kernel, and has the fastest convergence speed. We list the mean square deviation values in Table 1, which show that the algorithm achieves

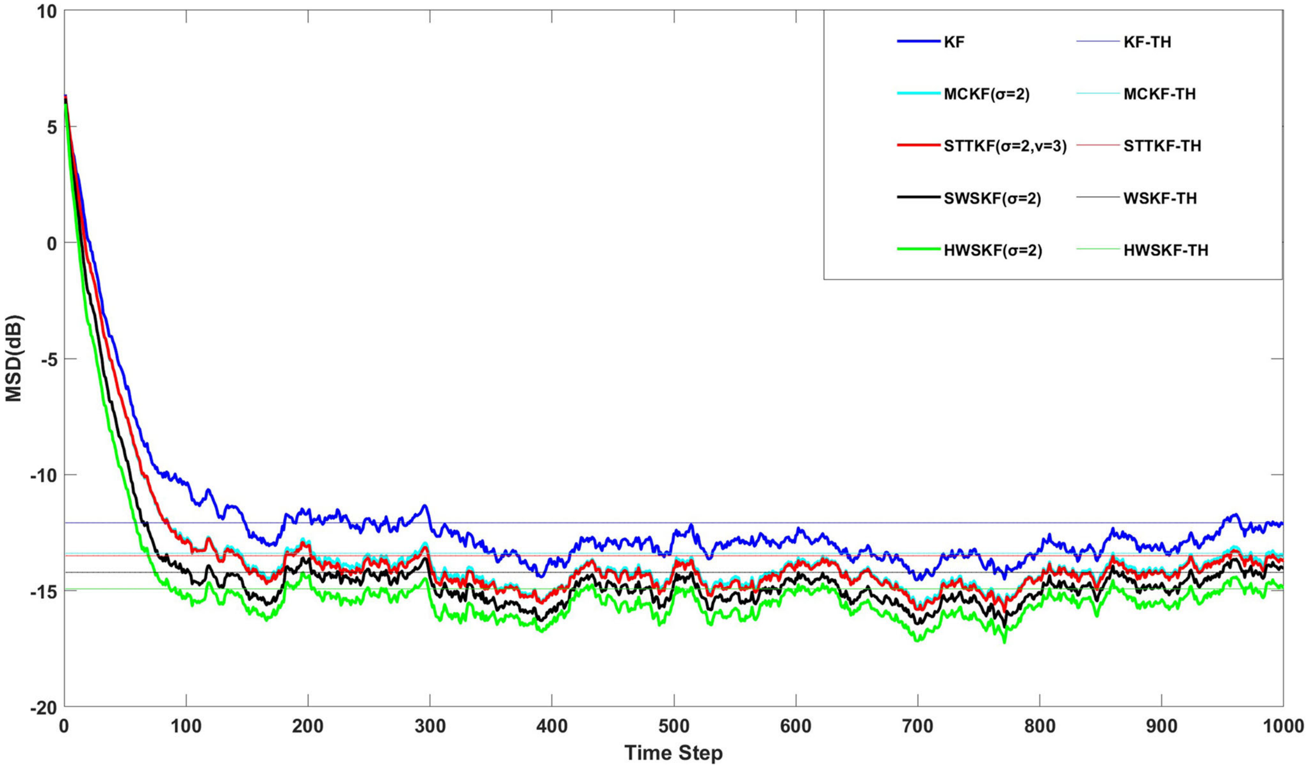


**Fig. 3** Comparison under impulse noise for case (2) in Example 1

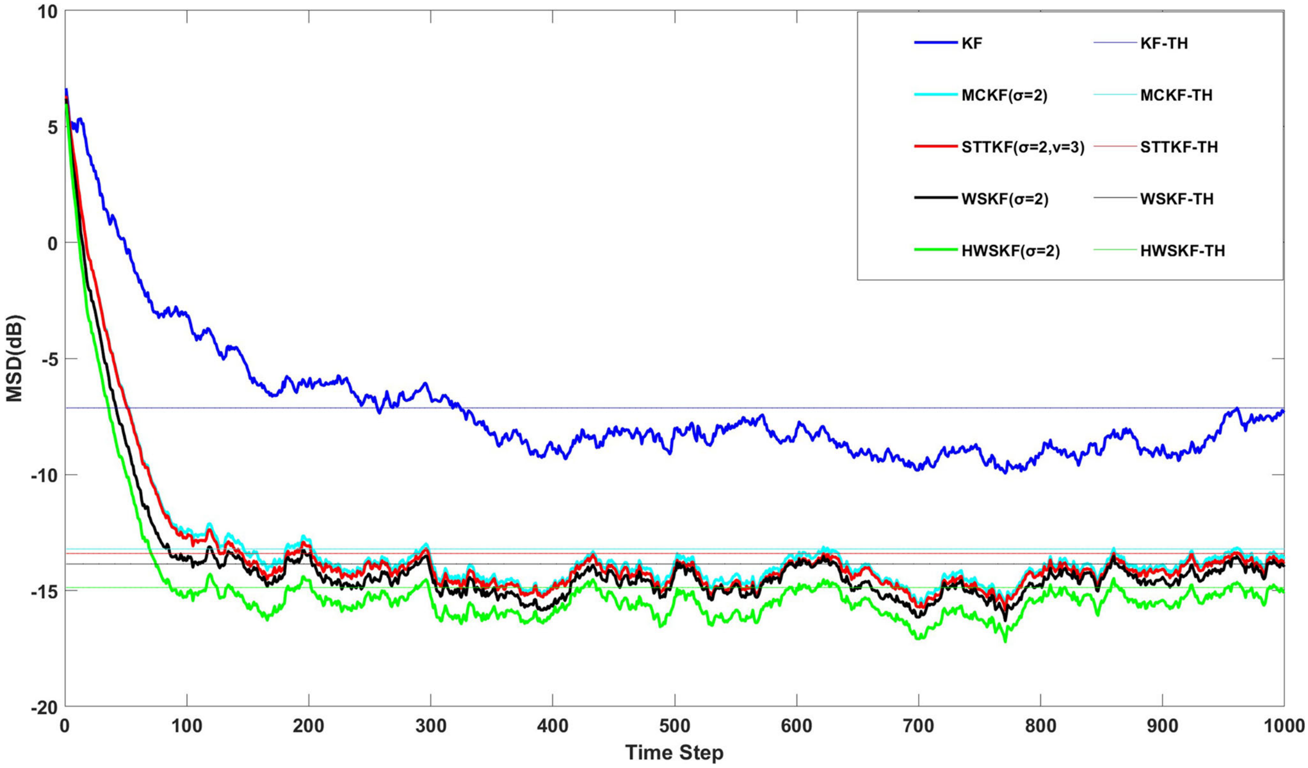


**Fig. 4** Comparison under impulse noise for case (3) in Example 1

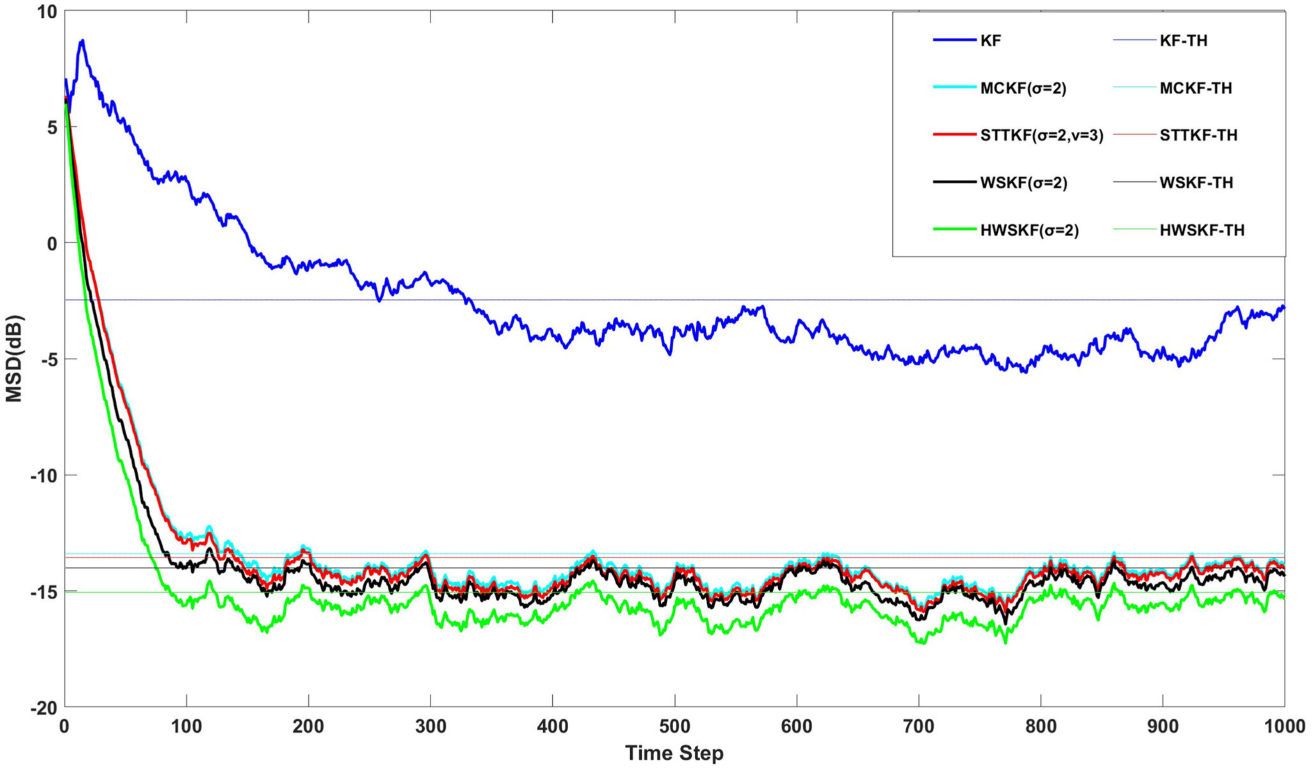


**Fig. 5** Comparison under impulse noise for case (4) in Example 1

**Table 2** MSDs (dB) of Example 1 under impulse noise

| | Case1 | Case2 | Case3 | Case4 |
|---|---|---|---|---|
| KF | −14.7330 | −12.0724 | −7.1217 | −2.4643 |
| MCKF ($\sigma = 2$) | −14.6940 | −13.3867 | −13.2001 | −13.3860 |
| STTKF ($\sigma = 2$, $v = 3$) | −14.6682 | −14.6682 | −13.4005 | −13.5625 |
| WSKF ($\sigma = 2$) | −14.7238 | −15.1467 | −13.8534 | −14.0083 |
| HWSKF ($\sigma = 2$) | −14.7238 | −15.5723 | −14.8701 | −15.0660 |
| MCKF ($\sigma = 3$) | −14.7158 | −13.0597 | −12.3741 | −12.4162 |
| STTKF ($\sigma = 3$, $v = 3$) | −14.7043 | −13.2361 | −12.7709 | −12.8398 |
| WSKF ($\sigma = 3$) | −14.7122 | −13.2687 | −12.2094 | −12.1267 |
| HWSKF ($\sigma = 3$) | −14.7122 | −14.4181 | −14.2341 | −14.3941 |
| MCKF ($\sigma = 5$) | −14.7268 | −12.6001 | −10.7658 | −10.1954 |
| STTKF ($\sigma = 5$, $v = 3$) | −14.7227 | −12.7856 | −11.4102 | −11.0481 |
| WSKF ($\sigma = 5$) | −14.7297 | −12.5695 | −10.0705 | −9.0950 |
| HWSKF ($\sigma = 5$) | −14.7297 | −14.1321 | −13.8165 | −13.8905 |

−14.9234, −14.9234 and 15.0660 in impulse noise($\sigma = 2$), which are all high steady state.

### 5.2 Simulation 2

In this section, we investigate the effect of the parameters of the HWSKF algorithm on the performance, such as kernel width $sigma$ and threshold $\iota$. The conclusions obtained can also guide the selection of the parameters of the HWSKF algorithm. This part is based on the conditional simulation of model 1.

First, we investigated the effect of kernel width $sigma$ on the performance with the kernel width set to $sigma =$ 0.5, 1, 2, 5, 10. The values of kernel width are shown in Table 3. The simulation plots for different $sigma$ are shown in Fig. 6. It can be obtained from them that when $sigma$ is near

**Table 3** MSD at different kernel widths *sigma*

| *sigma* | 0.5 | 1 | 2 | 5 | 10 |
|---|---|---|---|---|---|
| Case2 $\iota = 1e{-}4$ | −22.0129 | −17.4571 | −14.9234 | −15.0666 | −13.8905 |
| Case3 $\iota = 1e{-}4$ | −22.0698 | −17.3257 | −14.9234 | −13.8165 | −13.5766 |
| Case4 $\iota = 1e{-}4$ | −22.0695 | −13.4571 | −15.0666 | −13.8905 | −13.5458 |

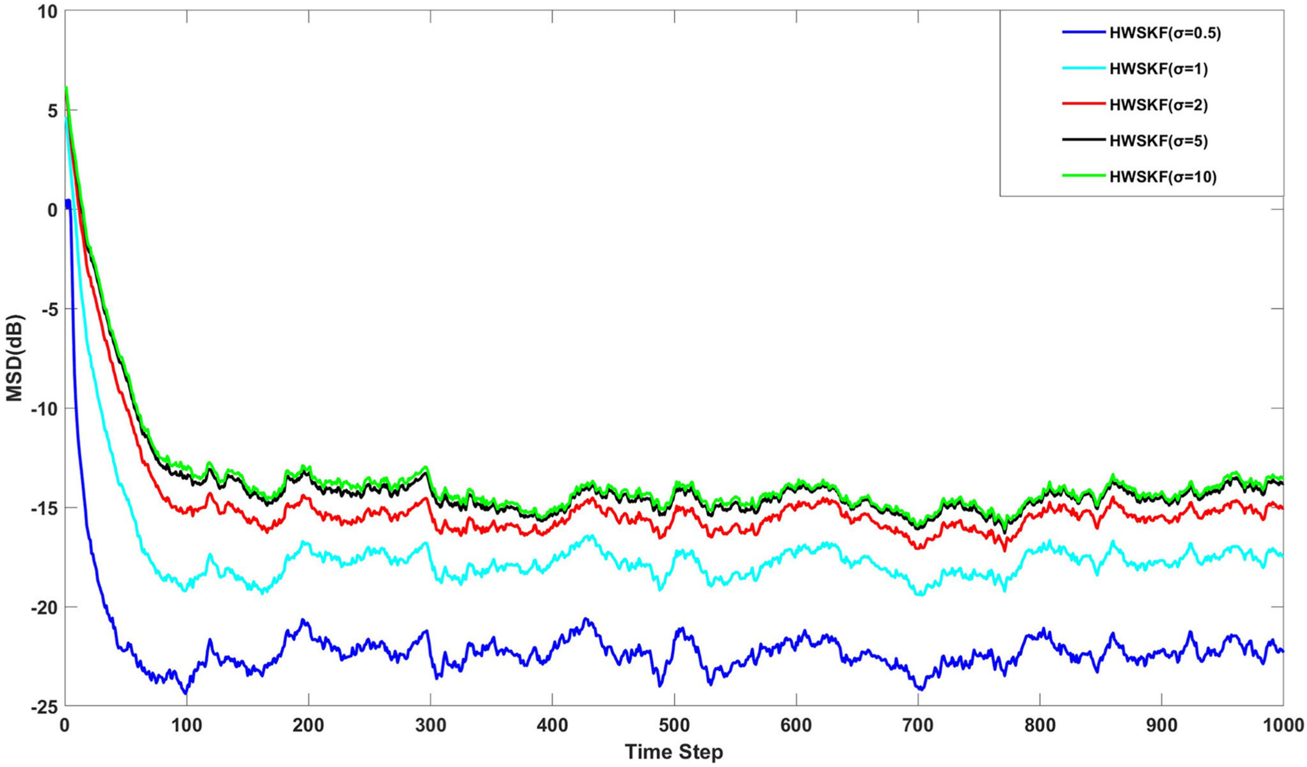


**Fig. 6** MSD maps for different core widths (case 3)

0.5, the MSD of the three impulse noises reaches the optimal results of −22.0129, −22.0689, and −22.0695, respectively, when the algorithm performance reaches near the optimal.

Then, a large number of experimental analyses were conducted in this paper at different thresholds $\iota$ (conducted in case 3), and the MSD plots and the table of MSD values at different thresholds are shown in Fig. 7 as well as in Table 4. We can see from this that the steady state performance of the algorithm increases as the threshold $\iota$ decreases. When the threshold $\iota$ is increased to 1e−4, the MSD reaches −14.8578 at this point, and when the threshold $\iota$ is increased to 1e−6, the MSD decreases by only −0.0113. Therefore, we choose $\iota = 1e{-}4$ for all thresholds to reduce the computational complexity of the algorithm, while ensuring that we have a high degree of steady state.

## 5.3 Simulation 3

Consider a land navigation model whose equations of state and observation can be modeled as

$$x_k = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1} \tag{54}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + v_k \tag{55}$$

where $t = 0.1$ s, the process noise and observation noise are the same as in simulation (1), and Gaussian noise and three kinds of impulse noise are used for the simulation process. In the experiments, the kernel width is taken as $\sigma = 2$ for MCKF, the parameters of STTKF are $\sigma = 2, v = 3$, the parameters of WSKF are $\sigma = 2$, and the choice of kernel width for HWSKF is the same as that for WSKF. The threshold is taken as $\iota = 1e{-}4$.

We compare different algorithms for fitting one of the two positional states of the simulation model in Fig. 8a, b. We find that the algorithm proposed in this paper is closer to the

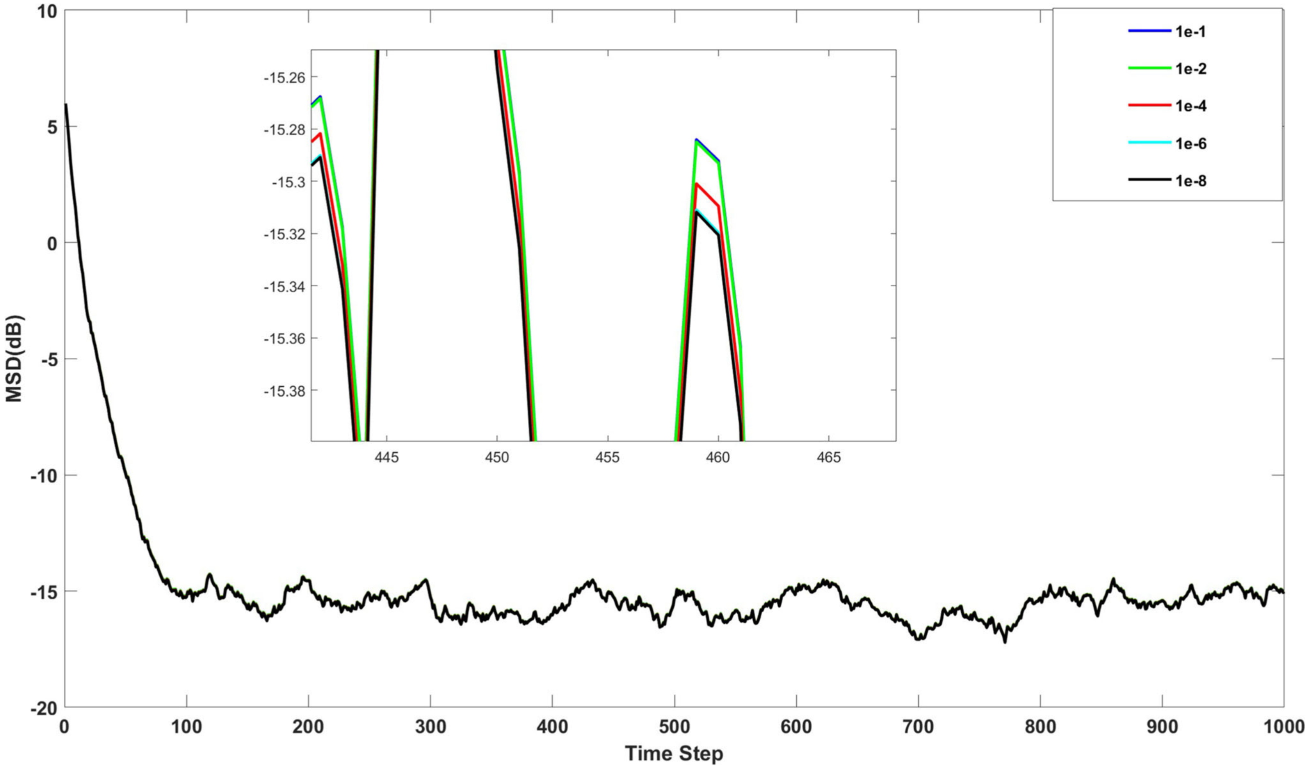


**Fig. 7** MSD graph at different thresholds (case 2)

**Table 4** MSD at different thresholds $\iota$

| Threshold | MSD |
|---|---|
| $\iota = 1e{-}1$ | $-13.5458$ |
| $\iota = 1e{-}2$ | $-14.8396$ |
| $\iota = 1e{-}4$ | $-14.8578$ |
| $\iota = 1e{-}6$ | $-14.8691$ |
| $\iota = 1e{-}8$ | $-14.8701$ |

real values than KF, MCKF, WSKF and STTKF, and has a better fitting effect. Similarly, Figs. 9, 10, and 11 show the performance of the algorithm in three different impulse noise environments, and the corresponding MSDs are presented in Table 5. We can clearly observe that the proposed algorithm in this paper possesses faster convergence speed and lower MSD compared to other algorithms under the influence of impulse noise, which are 8.7765, 9.1960 and 9.1960, respectively. Therefore, it can be shown that the algorithm in the second example also possesses better stationarity under the influence of impulse noise.

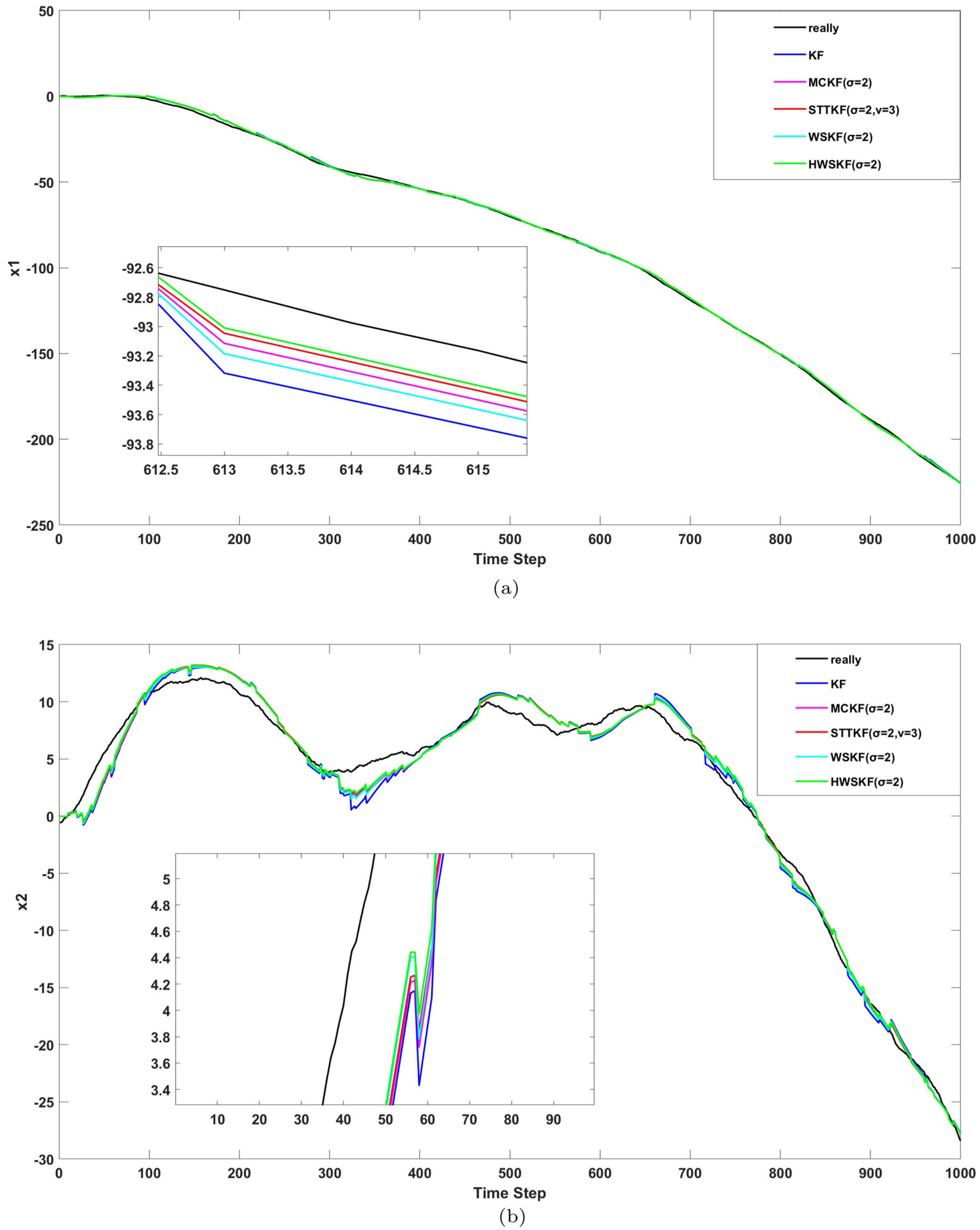


**Fig. 8** Position state (case 2). **a** Fitting plots for each algorithm at state 1; **b** fitting plots for each algorithm at state 2

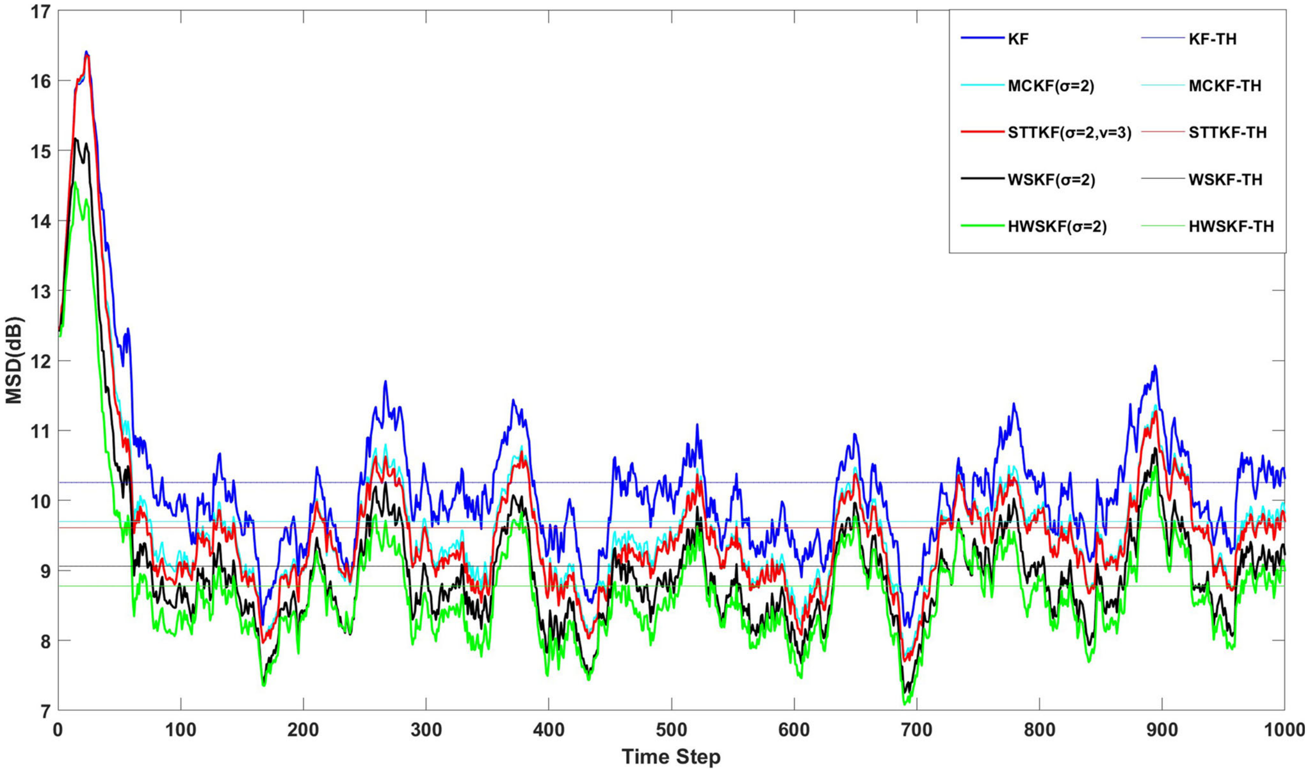


**Fig. 9** Comparison of case (2) in Example 2 under impulse noise

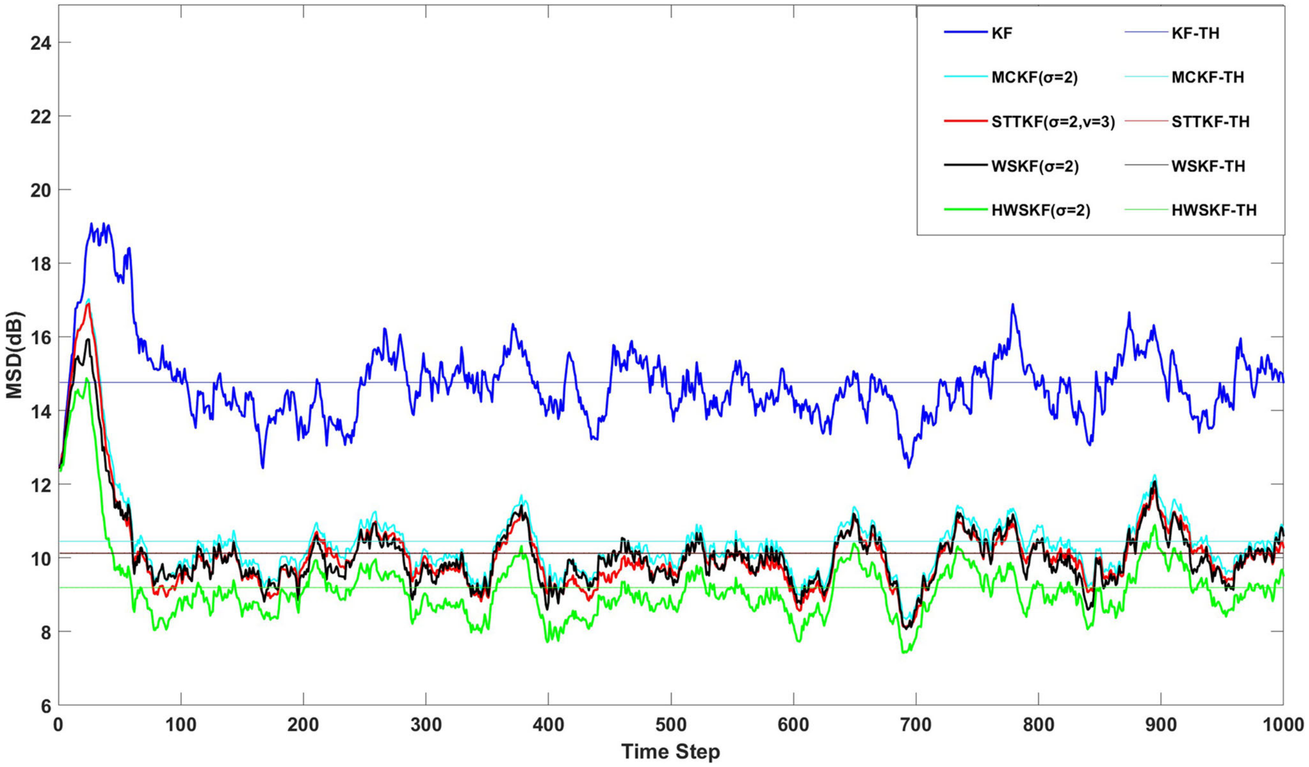


**Fig. 10** Comparison of case (3) in Example 2 under impulse noise

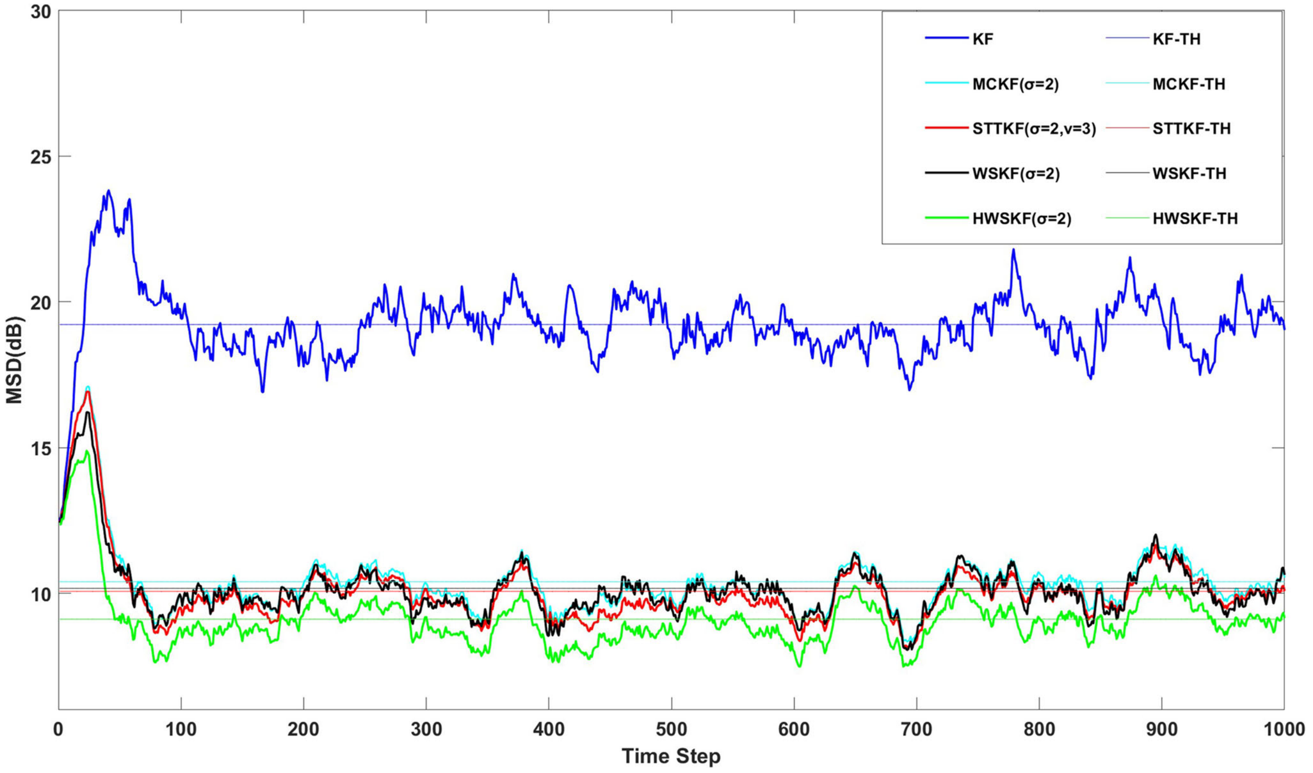


**Fig. 11** compares case (4) in Example 2 under impulse noise

**Table 5** MSDs (dB) of Example 1 under impulse noise

| | Case1 | Case2 | Case3 | Case4 |
|---|---|---|---|---|
| KF | 8.4212 | 11.2565 | 14.7621 | 19.2202 |
| MCKF ($\sigma = 2$) | 8.4845 | 9.6957 | 10.4514 | 10.3950 |
| STTKF ($\sigma = 2, v = 3$) | 8.5274 | 9.6102 | 10.1320 | 10.0686 |
| WSKF ($\sigma = 2$) | 8.4366 | 9.0592 | 10.1200 | 10.1696 |
| HWSKF ($\sigma = 2$) | 8.5649 | 8.7765 | 9.1960 | 9.1096 |

## 6 Conclusion

This paper proposes a Kalman filtering algorithm that fuses the S-function with the maximum correlation entropy under Huber estimated weights, the HSWKF algorithm. The algorithm utilizes the nonlinear properties of the function and the robust estimation of the weights to solve the problem of poor robustness of algorithms such as KF and MCKF in impulse noise environments. By simulation experiments, the stability of the algorithm is verified. The simulation results show that the experimental steady state errors of the four algorithms are consistent with the theoretical values, and the HWSKF proposed in this paper has the best steady state performance, which is better than the KF, MCKF, STTKF and WSKF in the future, the algorithm can be extended to the extended Kalman filtering algorithm and the estimation of the battery's charge state.

**Author Contributions** Y.H. was involved in the conceptualization, methodology, and software. K.Y. contributed to the algorithm design, equation derivation, Matlab simulation, and writing—reviewing and editing. Y.Q. performed the supervision. T.X. contributed to the software and validating writing—reviewing and editing.

**Data availability** The datasets analyzed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflicts of interest** The authors declare that they have no conflict of interest.

## References

1. Bai, Y.T., Yan, B., Zhou, C., Su, T., Jin, X.B.: State of art on state estimation: Kalman filter driven by machine learning. Ann. Rev. Control **56**, 100909 (2023). https://doi.org/10.1016/j.arcontrol.2023.100909
2. Lee, D., Vukovich, G., Lee, R.: Robust unscented Kalman filter for nanosat attitude estimation. Int. J. Control Autom. Syst. (2017). https://doi.org/10.1007/s12555-016-0498-4
3. Khalkhali, M., Vahedian, A., Sadoghi Yazdi, H.: Vehicle tracking with Kalman filter using online situation assessment. Robot.

Auton. Syst. **131**, 103596 (2020). https://doi.org/10.1016/j.robot.2020.103596

4. Lu, T., Watanabe, Y., Yamada, S., Takada, H.: Comparative evaluation of Kalman filters and motion models in vehicular state estimation and path prediction. J. Navig. (2021). https://doi.org/10.1017/S0373463321000370
5. Khalkhali, M., Vahedian, A., Sadoghi Yazdi, H.: Situation assessment-augmented interactive Kalman filter for multi-vehicle tracking. IEEE Trans. Intell. Transp. Syst. (2021). https://doi.org/10.1109/TITS.2021.3050878
6. Liu, X., Li, K., Wu, J., He, Y., Liu, X.: An extended Kalman filter based data-driven method for state of charge estimation of li-ion batteries. J. Energy Storage **40**, 102655 (2021). https://doi.org/10.1016/j.est.2021.102655
7. Chi, N.V., Vinh Thuy, N.: Soc estimation of the lithium-ion battery pack using a sigma point Kalman filter based on a cell's second order dynamic model. Appl. Sci. **10**, 1896 (2020). https://doi.org/10.3390/app10051896
8. Chaumette, E., Vilà-Valls, J., Vincent, F.: On the general conditions of existence for linear mmse filters: Wiener and Kalman. Signal Process. **184**, 108052 (2021). https://doi.org/10.1016/j.sigpro.2021.108052
9. Li, M., Tang, X., Zhang, Q., Zou, Y.: Non-gaussian pseudo-linear Kalman filtering-based target motion analysis with state constraints. Appl. Sci. **12**, 9975 (2022). https://doi.org/10.3390/app12199975
10. Ji, S., Kong, C., Sun, C., Zhang, J.F.: Kalman–Bucy filtering and minimum mean square estimator under uncertainty. SIAM J. Control Optim. **59**, 2669–2692 (2021). https://doi.org/10.1137/20M137954X
11. Pishdad, L., Labeau, F.: Analytic minimum mean-square error bounds in linear dynamic systems with gaussian mixture noise statistics. IEEE Access (2020). https://doi.org/10.1109/ACCESS.2020.2986420
12. Chen, B., Liu, X., Zhao, H., Principe, J.: Maximum correntropy Kalman filter. Automatica (2015). https://doi.org/10.1016/j.automatica.2016.10.004
13. Liu, C., Wang, G., Guan, X., Huang, C.: Robust m-estimation-based maximum correntropy Kalman filter. ISA Trans. (2022). https://doi.org/10.1016/j.isatra.2022.10.025
14. Wu, C., Hu, W., Meng, J., Xu, X., Huang, X., Cai, L.: State-of-charge estimation of lithium-ion batteries based on mcc-aekf in non-gaussian noise environment. Energy **274**, 127316 (2023). https://doi.org/10.1016/j.energy.2023.127316
15. Huang, H., Zhang, H.: Student's t-kernel-based maximum correntropy Kalman filter. Sensors **22**, 1683 (2022). https://doi.org/10.3390/s22041683
16. Wang, J., Lyu, D., He, Z., Zhou, H., Wang, D.: Cauchy kernel-based maximum correntropy Kalman filter. Int. J. Syst. Sci. **51**, 1–16 (2020). https://doi.org/10.1080/00207721.2020.1817614
17. Chen, B., Dang, L., Gu, Y., Zheng, N., Principe, J.: Minimum error entropy Kalman filter. IEEE Trans. Syst. Man Cybern. Syst. (2019). https://doi.org/10.1109/TSMC.2019.2957269
18. Yang, B., Cao, L., Ran, D., Xiao, B.: Centered error entropy Kalman filter with application to satellite attitude determination. Trans. Inst. Measur. Control **43**, 014233122110198 (2021). https://doi.org/10.1177/01423312211019867
19. Wang, H., Zhang, W., Zuo, J., Wang, H.: Outlier-robust Kalman filters with mixture correntropy. J. Frankl. Inst. (2020). https://doi.org/10.1016/j.jfranklin.2020.03.042
20. He, J., Wang, G., Yu, H., Liu, J., Peng, B.: Generalized minimum error entropy Kalman filter for non-gaussian noise. ISA Trans. (2022). https://doi.org/10.1016/j.isatra.2022.10.040
21. Ma, W., Kou, X., Xianzhi, H., Qi, A., Chen, B.: Recursive minimum kernel risk sensitive loss algorithm with adaptive gain factor for robust power system s estimation. Electr. Power Syst. Res. **206**, 107788 (2022). https://doi.org/10.1016/j.epsr.2022.107788
22. Lin, D., Zhang, Q., Chen, X., Wang, S.: Maximum correntropy quaternion Kalman filter. IEEE Trans. Signal Process. **71**, 2792–2803 (2023). https://doi.org/10.1109/TSP.2023.3300631
23. You, K., Xie, L.: Kalman filtering with scheduled measurements. IEEE Trans. Signal Process. **61**, 1520–1530 (2012). https://doi.org/10.1109/TSP.2012.2235436
24. Wenkang, W., Jingan, F., Bao, S., Xinxin, L.: Huber-based robust unscented Kalman filter distributed drive electric vehicle state observation. Energies **14**, 750 (2021). https://doi.org/10.3390/en14030750
25. Xu, W., Zhao, H., Zhou, L.: Modified Huber m-estimate function-based distributed constrained adaptive filtering algorithm over sensor network. IEEE Sens. J. (2022). https://doi.org/10.1109/JSEN.2022.3201584
26. Hou, J., He, H., Yang, Y., Gao, T., Zhang, Y.: A variational Bayesian and Huber-based robust square root cubature Kalman filter for lithium-ion battery state of charge estimation. Energies **12**, 1717 (2019). https://doi.org/10.3390/en12091717

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.